



# Upgrading to Certified Linux Engineer 10

COURSE 3076

**Novell Training Services**

[www.novell.com](http://www.novell.com)

AUTHORIZED COURSEWARE

---

## Proprietary Statement

Copyright © 2006 Novell, Inc. All rights reserved.

No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express prior consent of the publisher. This manual, and any portion thereof, may not be copied without the express written permission of Novell, Inc. Novell, Inc.

1800 South Novell Place  
Provo, UT 84606-2399

## Disclaimer

Novell, Inc. makes no representations or warranties with respect to the contents or use of this manual, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

Further, Novell, Inc. reserves the right to revise this publication and to make changes in its content at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any NetWare software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

Further, Novell, Inc. reserves the right to make changes to any and all parts of NetWare software at any time, without obligation to notify any person or entity of such changes.

This Novell Training Manual is published solely to instruct students in the use of Novell networking software. Although third-party application software packages are used in Novell training courses, this is for demonstration purposes only and shall not constitute an endorsement of any of these software applications.

Further, Novell, Inc. does not represent itself as having any particular expertise in these application software packages and any use by students of the same shall be done at the students' own risk.

## Software Piracy

Throughout the world, unauthorized duplication of software is subject to both criminal and civil penalties.

If you know of illegal copying of software, contact your local Software Antipiracy Hotline.

For the Hotline number for your area, access Novell's World Wide Web page at <http://www.novell.com> and look for the piracy page under "Programs."

Or, contact Novell's anti-piracy headquarters in the U.S. at 800-PIRATES (747-2837) or 801-861-7101.

## Trademarks

Novell, Inc. has attempted to supply trademark information about company names, products, and services mentioned in this manual. The following list of trademarks was derived from various sources.

### Novell, Inc. Trademarks

Novell, the Novell logo, NetWare, BorderManager, ConsoleOne, DirXML, GroupWise, iChain, ManageWise, NDPS, NDS, NetMail, Novell Directory Services, Novell iFolder, Novell SecretStore, Ximian, Ximian Evolution and ZENworks are registered trademarks; CDE, Certified Directory Engineer and CNE are registered service marks; eDirectory, Evolution, exteNd, exteNd Composer, exteNd Directory, exteNd Workbench, Mono, NIMS, NLM, NMAS, Novell Certificate Server, Novell Client, Novell Cluster Services, Novell Distributed Print Services, Novell Internet Messaging System, Novell Storage Services, Nsure, Nsure Resources, Nterprise, Nterprise Branch Office, Red Carpet and Red Carpet Enterprise are trademarks; and Certified Novell Administrator, CNA, Certified Novell Engineer, Certified Novell Instructor, CNI, Master CNE, Master CNI, MCNE, MCNI, Novell Education Academic Partner, NEAP, Ngage, Novell Online Training Provider, NOTP and Novell Technical Services are service marks of Novell, Inc. in the United States and other countries. SUSE is a registered trademark of SUSE LINUX GmbH, a Novell company. For more information on Novell trademarks, please visit <http://www.novell.com/company/legal/trademarks/tmlist.html>.

### Other Trademarks

Adaptec is a registered trademark of Adaptec, Inc. AMD is a trademark of Advanced Micro Devices. AppleShare and AppleTalk are registered trademarks of Apple Computer, Inc. ARCServ is a registered trademark of Cheyenne Software, Inc. Btrieve is a registered trademark of Pervasive Software, Inc. EtherTalk is a registered trademark of Apple Computer, Inc. Java is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries. Linux is a registered trademark of Linus Torvalds. LocalTalk is a registered trademark of Apple Computer, Inc. Lotus Notes is a registered trademark of Lotus Development Corporation. Macintosh is a registered trademark of Apple Computer, Inc. Netscape Communicator is a trademark of Netscape Communications Corporation. Netscape Navigator is a registered trademark of Netscape Communications Corporation. Pentium is a registered trademark of Intel Corporation. Solaris is a registered trademark of Sun Microsystems, Inc. The Norton AntiVirus is a trademark of Symantec Corporation. TokenTalk is a registered trademark of Apple Computer, Inc. Tru64 is a trademark of Digital Equipment Corp. UnitedLinux is a registered trademark of UnitedLinux. UNIX is a registered trademark of the Open Group. WebSphere is a trademark of International Business Machines Corporation. Windows and Windows NT are registered trademarks of Microsoft Corporation.

All other third-party trademarks are the property of their respective owners.

# Contents

## Introduction

Course Objectives .....	Intro-2
Audience .....	Intro-2
Certification and Prerequisites .....	Intro-3
SUSE Linux Enterprise Server 10 Support and Maintenance .....	Intro-5
Novell Customer Center .....	Intro-6
SUSE Linux Enterprise Server 10 Online Resources .....	Intro-7
Exercise Conventions .....	Intro-8

## SECTION 1 Installation of SUSE Linux Enterprise Server 10

Objective .....	1-1
Objective 1 Understand the Novell Customer Center Configuration and Online Update .....	1-2
Summary .....	1-5

## SECTION 2 Use the GNOME Desktop Environment

Objectives .....	2-1
Objective 1 Log In .....	2-2
Objective 2 Log Out and Shut Down .....	2-4
Objective 3 Identify GNOME Desktop Components .....	2-6

Objective 4	Manage Icons in GNOME . . . . .	2-10
	Desktop . . . . .	2-10
	Panel . . . . .	2-13
	Main Menu . . . . .	2-14
Objective 5	Use the GNOME File Manager (Nautilus) . . . . .	2-15
Objective 6	Search for Files . . . . .	2-17
Objective 7	Access the Command Line Interface From the Desktop . . . . .	2-19
	Summary . . . . .	2-20

## **SECTION 3   Manage Hardware**

	Objectives . . . . .	3-1
Objective 1	Describe the Differences between Devices and Interfaces . . .	3-2
Objective 2	Describe how Device Drivers Work . . . . .	3-3
Objective 3	Describe how Device Drivers Are Loaded . . . . .	3-6
Objective 4	Manage Kernel Modules Manually . . . . .	3-7
	Kernel Module Basics . . . . .	3-7
	Manage Modules from the Command Line . . . . .	3-8
	modprobe Configuration File (/etc/modprobe.conf) . . . . .	3-11
	<b>Exercise 3-1</b> Manage the Linux Kernel Modules . . . . .	3-12
Objective 5	Describe the sysfs File System . . . . .	3-14
Objective 6	Describe how udev Works . . . . .	3-17
	Understand the Purpose of udev . . . . .	3-17
	Understand how udev Works . . . . .	3-18
	Understand Persistent Interface Names . . . . .	3-19
	<b>Exercise 3-2</b> Add a device symlink with udev . . . . .	3-22
Objective 7	Use the hwup Command . . . . .	3-23
	From Configuration Files . . . . .	3-24
	From sysfs . . . . .	3-27

	<b>Exercise 3-3 Explore Hardware Initialization</b> . . . . .	3-28
	Summary . . . . .	3-30
<b>SECTION 4</b>	<b>Configure Linux File System Partitions</b>	
	Objectives . . . . .	4-1
Objective 1	Finalize Partitioning . . . . .	4-2
Objective 2	Configure LVM with Command Line Tools . . . . .	4-3
	Tools to Administer Physical Volumes . . . . .	4-3
	Tools to Administer Volume Groups . . . . .	4-4
	Tools to Administer Logical Volumes . . . . .	4-5
	Summary . . . . .	4-6
<b>SECTION 5</b>	<b>Use the NetworkManager to Configure the Network</b>	
	Objective . . . . .	5-1
Objective 1	Use the NetworkManager to Configure the Network . . . . .	5-2
	Summary . . . . .	5-5
<b>SECTION 6</b>	<b>Administer User Access and Security</b>	
	Objectives . . . . .	6-1
Objective 1	Configure User Authentication with PAM . . . . .	6-2
	Location and Purpose of PAM Configuration Files . . . . .	6-4
	PAM Configuration . . . . .	6-5
	PAM Configuration File Examples . . . . .	6-8
	Secure Password Guidelines . . . . .	6-11
	PAM Documentation Resources . . . . .	6-12
	<b>Exercise 6-1</b> Configure PAM Authentication . . . . .	6-13
Objective 2	Configure Security Settings . . . . .	6-16

<b>Exercise 6-2</b> Configure the Password Security Settings . . . . .	6-27
Summary . . . . .	6-29

## **SECTION 7    Use Syslog Daemon syslog-ng**

Objectives . . . . .	7-1
Objective 1    Use Syslog Daemon syslog-ng . . . . .	7-2
/etc/sysconfig/syslog . . . . .	7-3
/etc/syslog-ng/syslog-ng.conf.in . . . . .	7-4
/etc/syslog-ng/syslog-ng.conf . . . . .	7-4
Summary . . . . .	7-11

## **SECTION 8    Manage Virtualization with Xen**

Objectives . . . . .	8-1
Objective 1    Understand the Concept of Virtualization . . . . .	8-2
Objective 2    Understand How Xen Works . . . . .	8-3
Understand Virtualization Methods . . . . .	8-4
Understand the Xen Architecture . . . . .	8-6
Objective 3    Install Xen . . . . .	8-8
<b>Exercise 8-1</b> Install Xen . . . . .	8-11
Objective 4    Manage Xen Domains with YaST . . . . .	8-13
<b>Exercise 8-2</b> Install a Guest Domain . . . . .	8-20
Objective 5    Manage Xen Domains at the Command Line . . . . .	8-22
Understand a Domain Configuration File . . . . .	8-22
Use the xm Tool . . . . .	8-24
<b>Exercise 8-3</b> Change Memory Allocation of a Guest Domain . . . . .	8-28
Automate Domain Startup and Shutdown . . . . .	8-30
<b>Exercise 8-4</b> Automate Domain Startup . . . . .	8-31

Objective 6	Understand Xen Networking . . . . .	8-32
	Understand the Basic Networking Concept . . . . .	8-32
	Understand Bridging . . . . .	8-33
	Understand the Network Interfaces in domain0 . . . . .	8-34
	<b>Exercise 8-5</b> Check the Network Configuration . . . . .	8-38
Objective 7	Migrate a Guest Domain . . . . .	8-39
	Use Domain Save and Restore . . . . .	8-39
	Use Migration and Live Migration . . . . .	8-40
	Summary . . . . .	8-41
 <b>SECTION 9     Configure a DNS Server Using BIND</b>		
	Objectives . . . . .	9-1
Objective 1	Create a Key for Zone Transfer . . . . .	9-2
Objective 2	Configure Dynamic DNS . . . . .	9-4
	Summary . . . . .	9-8
 <b>SECTION 10    Configure DHCP Pools and Failover</b>		
	Objectives . . . . .	10-1
Objective 1	Configure DHCP Pools . . . . .	10-2
Objective 2	Configure DHCP Failover . . . . .	10-4
	Basics of DHCP Failover . . . . .	10-4
	Configure Failover . . . . .	10-5
	Summary . . . . .	10-19
 <b>SECTION 11    Manage OpenLDAP</b>		
	Objectives . . . . .	11-1

Objective 1	Install and Set Up an OpenLDAP Server . . . . .	11-2
	Install the Required Software and Start the Server . . . . .	11-2
	Configure OpenLDAP with YaST . . . . .	11-6
	<b>Exercise 11-1</b> Set Up OpenLDAP with YaST . . . . .	11-21
	Edit the OpenLDAP Configuration Files . . . . .	11-24
Objective 2	Activate LDAP Authentication . . . . .	11-34
	Change the User Password . . . . .	11-34
	Activate pam_ldap . . . . .	11-37
	<b>Exercise 11-2</b> Set up an LDAP User Database . . . . .	11-39
Objective 3	Replicate OpenLDAP Servers . . . . .	11-43
	Add the Replicaton DN to the LDAP Directory . . . . .	11-43
	Configure slapd for Replication . . . . .	11-44
	The Command-Line Options of slurpd . . . . .	11-46
	Transfer the LDAP Database . . . . .	11-47
	<b>Exercise 11-3</b> Replicate OpenLDAP Servers . . . . .	11-48
	Summary . . . . .	11-52

## **SECTION 12    Configure a Mail Server**

	Objectives . . . . .	12-1
Objective 1	Understand SMTP Communication. . . . .	12-2
	The SMTP Commands . . . . .	12-2
	Command Syntax . . . . .	12-5
	SMTP Reply Codes . . . . .	12-6
	Minimal SMTP Command Implementation . . . . .	12-8
	An Example for Sending Mail with Telnet . . . . .	12-9
Objective 2	Manage Spam. . . . .	12-10
	Use SpamAssassin . . . . .	12-10
	Test SpamAssassin . . . . .	12-11
Objective 3	Use a Virus Scanner for Email. . . . .	12-12
	AVMailGate . . . . .	12-12



---

	<b>Exercise 12-1</b> Use AVMailGate as a Virus Scanner for Email . . . . .	12-24
	AMaViSd-new . . . . .	12-27
	<b>Exercise 12-2</b> Use AMaViSd as Virus Scanner for Email . .	12-40
	Summary . . . . .	12-44
<b>SECTION 13</b>	<b>Apply Security</b>	
	Objectives . . . . .	13-1
Objective 1	Apply Security Updates . . . . .	13-2
	Configure the Novell Customer Center . . . . .	13-3
	Use the YaST Online Update . . . . .	13-4
Objective 2	Understand Recent Match of iptables . . . . .	13-7
Objective 3	Log to a Remote Host . . . . .	13-8
	Client Side Configuration of Syslog-ng . . . . .	13-8
	Server Side Configuration of Syslog-ng . . . . .	13-9
	<b>Exercise 13-1</b> Log to a Remote Host . . . . .	13-10
	Summary . . . . .	13-13
<b>SECTION 14</b>	<b>AppArmor</b>	
	Objectives . . . . .	14-1
Objective 1	Improve Application Security with AppArmor . . . . .	14-2
Objective 2	Create and Manage AppArmor Profiles . . . . .	14-4
	Understand Profiles and Rules . . . . .	14-5
	Administer AppArmor Profiles with YaST . . . . .	14-8
	Administer AppArmor Profiles with Command Line Tools . . . . .	14-17
	<b>Exercise 14-1</b> AppArmor . . . . .	14-22

Objective 3	Control AppArmor . . . . .	14-27
	Start and Stop AppArmor . . . . .	14-27
	View AppArmor's Status . . . . .	14-28
	Reload Profiles . . . . .	14-31
Objective 4	Monitor AppArmor . . . . .	14-32
	Security Event Report . . . . .	14-32
	Security Event Notification . . . . .	14-35
	Summary . . . . .	14-37

# Introduction

The *Upgrading to Certified Linux Engineer 10* (3076) course is designed to provide information on the differences between SUSE Linux Enterprise Server version 9 and version 10 related especially to the topics covered in the Novell Certified Linux Engineer (CLE) curriculum:

- *SUSE Linux Enterprise Server 10 Fundamentals* (Course 3071)
- *SUSE Linux Enterprise Server 10 Administration* (Course 3072)
- *SUSE Linux Enterprise Server 10 Advanced Administration* (Course 3073)
- *SUSE Linux Enterprise Server 10: Networking Services* (Course 3074)
- *SUSE Linux Enterprise Server 10: Security* (Course 3075)

You will learn about the new features as well as the changes in SUSE Linux Enterprise Server 10 compared to your existing knowledge as a Novell Certified Linux Engineer.

The course is available as a learning module through the Internet, which takes approximately 8 hours to complete. In order to gain sufficient experience to take the new Novell Certified Linux Engineer 10 (CLE 10) Practicum Exam.

Novell highly recommends that you register with one of the Novell Authorised Training Partners globally for an Upgrade Workshop. Based on this course the content is enriched with the option to take the new Practicum Exam at the end of the class.

## Course Objectives

This course teaches you the following concepts and skills to understanding the differences to SUSE Linux Enterprise Server 10:

- Installation of SUSE Linux Enterprise Server 10
- Use the GNOME Desktop Environment
- Manage Hardware
- Configure Linux File System Partitions
- Use the NetworkManager to Configure the Network
- Administer User Access and Security
- Use Syslog Daemon syslog-ng
- Manage Virtualization with Xen
- Configure a DNS Server Using BIND
- Configure DHCP Pools and Failover
- Manage OpenLDAP
- Configure a Mail Server
- Apply Security
- AppArmor

## Audience

This course is designed for Novell Certified Linux Engineers or those with equivalent knowledge. This course content combined with the Upgrade Workshops offered through Novell's Training Partners globally is the ideal preparation in order to take the Novell Certified Linux Engineer 10 (CLE 10) Practicum Exam.

## Certification and Prerequisites

This course helps you prepare for the Novell Certified Linux Engineer (Novell CLE 10) Practical Test, called a practicum. The Novell CLE 10 is a higher-level certification for people who passed the Novell CLP Practicum.

As with all Novell certifications, course work is recommended. To achieve the certification, you are required to pass the Novell CLE 10 Practicum (050–698).

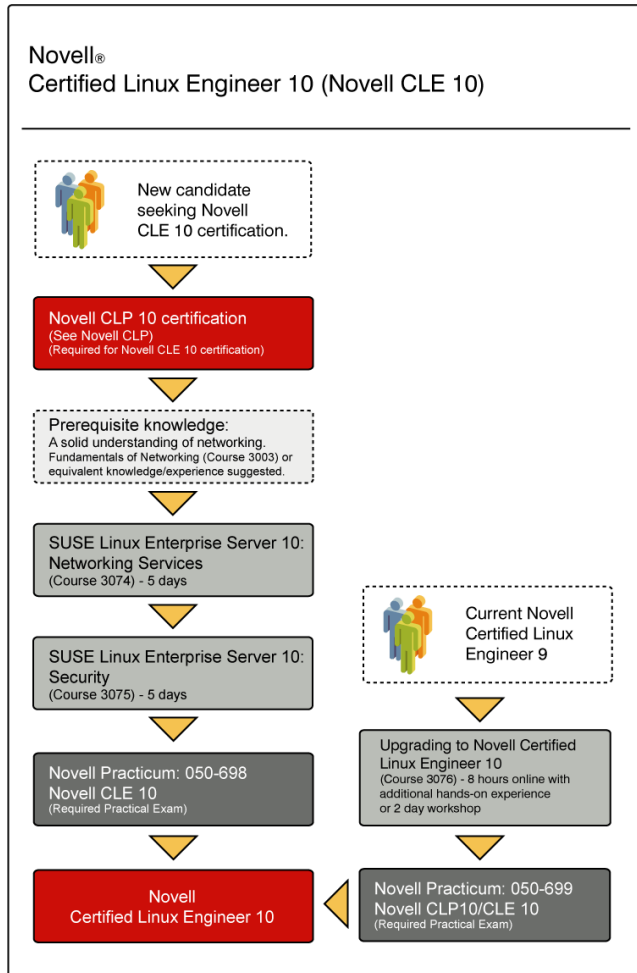
The Novell CLP Practicum is a hands-on, scenario-based exam where you apply the knowledge you have learned to solve real-life problems—demonstrating that you know what to do and how to do it.

The practicum tests you on objectives in the following Novell CLE 10 courses:

- *SUSE Linux Enterprise Server 10: Networking Services* (Course 3074)
- *SUSE Linux Enterprise Server 10: Security* (Course 3075)

The following illustrates the training/testing path for Novell CLE 10:

**Figure Intro-1 .**





---

For more information about Novell certification programs and taking the Novell CLP 10 and CLE 10 Practicum exam, see <http://www.novell.com/training/certinfo/>.

---

## **SUSE Linux Enterprise Server 10 Support and Maintenance**

To receive official support and maintenance updates for SUSE Linux Enterprise Server 10, you need to do one of the following:

- Register for a free registration/serial code that provides you with 30 days of support and maintenance.
- Purchase a copy of SUSE Linux Enterprise Server 10 from Novell (or an authorized dealer).

You can obtain your free 30-day support and maintenance code at <http://www.novell.com/products/linuxenterpriseserver/eval.html>.



---

You will need to have or create a Novell login account to access the 30-day evaluation.

---

## Novell Customer Center

Novell Customer Center is an intuitive, web-based interface that helps you to manage your business and technical interactions with Novell. Novell Customer Center consolidates access to information, tools and services such as:

- Automated registration for new SUSE Linux Enterprise products
- Patches and updates for all shipping Linux products from Novell
- Order history for all Novell products, subscriptions and services
- Entitlement visibility for new SUSE Linux Enterprise products
- Linux subscription-renewal status
- Subscription renewals via partners or Novell

For example, a company might have an administrator who needs to download SUSE Linux Enterprise software updates, a purchaser who wants to review the order history and an IT manager who has to reconcile licensing. With Novell Customer Center, the company can meet all these needs in one location and can give each user access rights appropriate to their roles.

You can access the Novell Customer Center at <http://www.novell.com/center>.



## SUSE Linux Enterprise Server 10 Online Resources

Novell provides a variety of online resources to help you configure and implement SUSE Linux Enterprise Server 10.

These include the following:

- <http://www.novell.com/products/linuxenterpriseserver/>  
This is the Novell home page for SUSE Linux Enterprise Server 10.
- <http://www.novell.com/documentation/sles10/index.html>  
This is the Novell Documentation web site for SUSE Linux Enterprise Server 10.
- <http://support.novell.com/linux/>  
This is the home page for all Novell Linux support, and includes links to support options such as the Knowledgebase, downloads, and FAQs.
- <http://www.novell.com/coolsolutions>  
This Novell web site provides the latest implementation guidelines and suggestions from Novell on a variety of products, including SUSE Linux.

## Exercise Conventions

When working through an exercise, you will see conventions that indicate information you need to enter that is specific to your server.

The following describes the most common conventions:

- ***italicized/bolded text***. This is a reference to your unique situation, such as the host name of your server.

For example, if the host name of your server is DA50, and you see the following,

***hostname.digitalairlines.com***

you would enter

**DA50.digitalairlines.com**

- **10.0.0.xx**. This is the IP address that is assigned to your SUSE Linux Enterprise Server 10 server.

For example, if your IP address is 10.0.0.50, and you see the following

**10.0.0.xx**

you would enter

**10.0.0.50**

- **Select**. The word *select* is used in exercise steps to indicate a variety of actions including clicking a button on the interface and selecting a menu item.
- **Enter and Type**. The words *enter* and *type* have distinct meanings.

The word *enter* means to type text in a field or at a command line and press the Enter key when necessary. The word *type* means to type text without pressing the Enter key.

If you are directed to type a value, make sure you do not press the Enter key or you might activate a process that you are not ready to start.

## **SECTION 1**    **Installation of SUSE Linux Enterprise Server 10**

There are little changes in the installation process of SUSE Linux Enterprise Server 9 to SUSE Linux Enterprise Server 10.

In SUSE Linux Enterprise Server 9 the default authentication method was using an LDAP server. On SUSE Linux Enterprise Server 10 it is local authentication using the file `/etc/passwd`.

Another new feature is the Novell Customer Center.

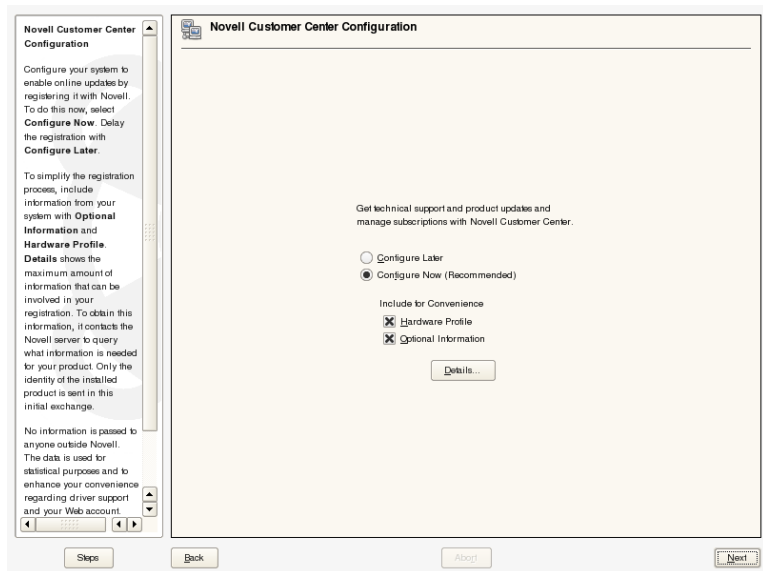
### **Objective**

1. Understand the Novell Customer Center Configuration and Online Update

## Objective 1 Understand the Novell Customer Center Configuration and Online Update

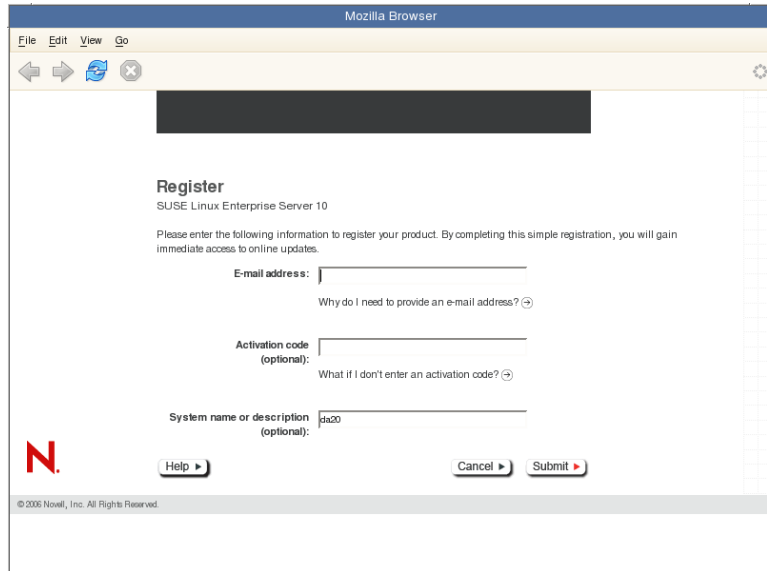
If the Internet connection test was successful, you can configure the Novell Customer Center, which is required to perform an online update. If there are any update packages available on the SUSE update servers, you can download and install them to fix known bugs or security issues.

Figure 1-1



Selecting **Next** starts a Browser and connects to the Novell web site, where all you have to enter is your e-mail address, and an activation code, if available.

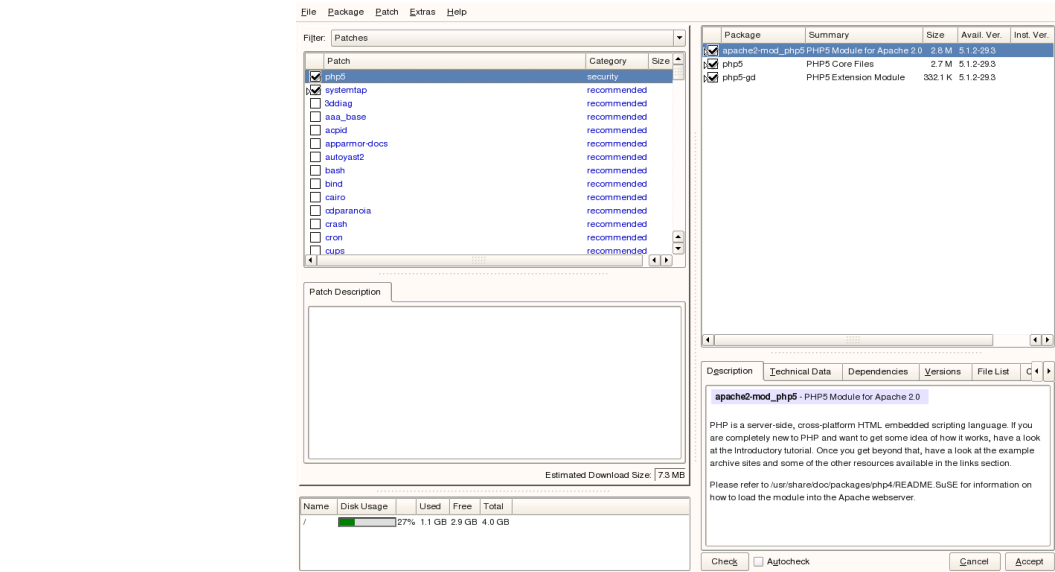
**Figure 1-2**



After successful registration, the Online Update dialog opens. You can start the Online Update by selecting **Run Update** and **Next**. (You can also select **Skip Update** to perform the update later in the installed system.)

YaST's online update dialog opens up with a list of available patches (if any).

Figure 1-3



Select the patches you want to install, and then start the update process by selecting **Accept**.

Once the installation is complete, visit the Novell Customer Center at <http://www.novell.com/center/> to administer your Novell products and subscriptions.



In the **Filter** pull-down menu of the software selection dialog there is an item **Patterns**. This filter displays all software that is available on the known installation media. It is grouped in predefined sets of packages that logically belong together. In SUSE Linux Enterprise Server 10 this item was labeled **Selections**.

## Summary

Objective	Summary
1. Understand the Novell Customer Center Configuration and Online Update	<p>On SUSE Linux Enterprise Server 10 it is local authentication using the file <code>/etc/passwd</code>.</p> <p>The Novell Customer Center is required to perform an online update.</p>

---





## SECTION 2    Use the GNOME Desktop Environment

GNOME is a comfortable desktop environment. GNOME supports drag and drop. Numerous programs are specifically designed for GNOME.

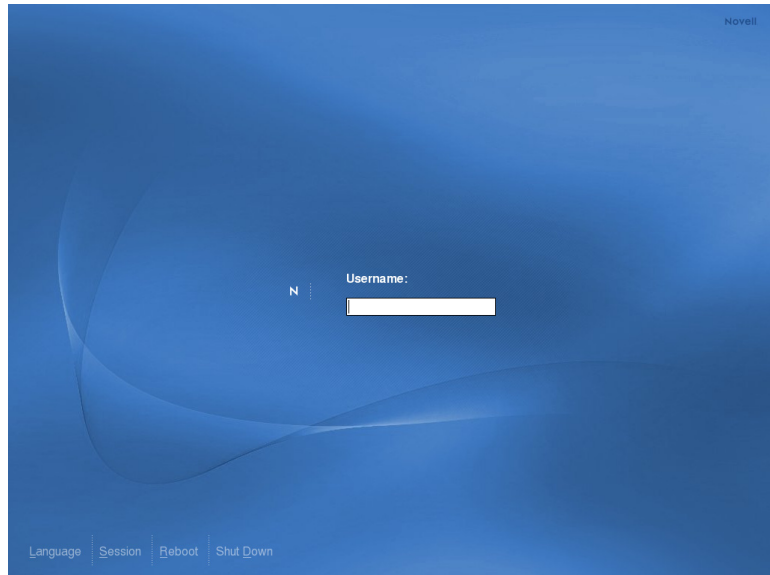
### Objectives

1. Log In
2. Log Out and Shut Down
3. Identify GNOME Desktop Components
4. Manage Icons in GNOME
5. Use the GNOME File Manager (Nautilus)
6. Search for Files
7. Access the Command Line Interface From the Desktop

## Objective 1    Log In

When the computer is booted and ready for work, the following login dialog appears:

**Figure 2-1**



After entering a username, press **Enter**. Then enter your password and press **Enter** again. If the login is successful, the following GNOME desktop environment appears:

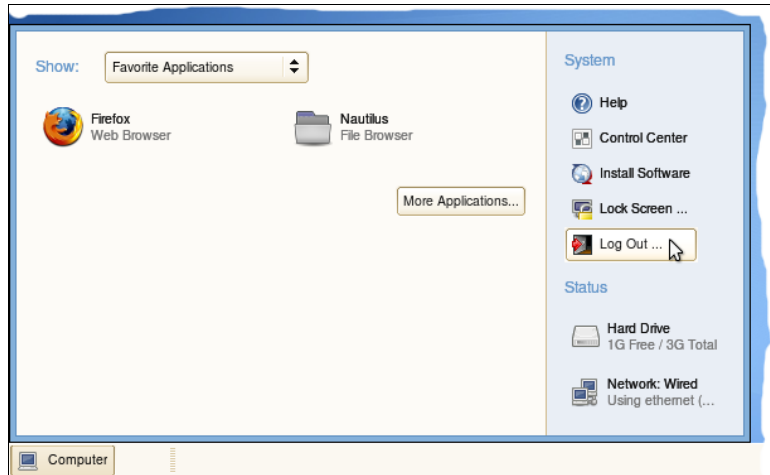
**Figure 2-2**



## Objective 2 Log Out and Shut Down

When you are ready to log out of the system, open the **Computer** menu (also called *main menu*) in the bottom panel.

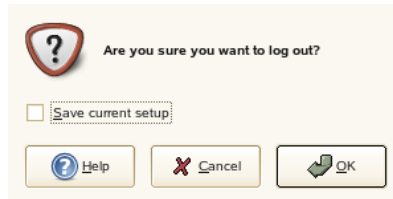
**Figure 2-3**



At the right side of the **Computer** menu, select the **Log Out** entry.

After selecting **Log Out**, a confirmation dialog appears.

**Figure 2-4**



If you select **Save current setup**, your current desktop environment settings are saved and restored after your next login.

Select **OK** after selecting an action.

If you are at the login screen, there are four options available in the lower left corner:

- **Language.** Select the language of the desktop environment.
- **Session.** You can choose a window manager other than GNOME. In this student manual, we cover only GNOME (the default window manager). Some basic informations about the KDE environment you find in the appendix A.
- **Reboot.** Reboots the system.



---

Only root is allowed to reboot the system. So you have to enter the root password.

---

- **Shut Down.** Shuts down your computer.



---

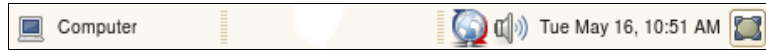
Only root is allowed to shut down the system. So you have to enter the root password.

---

## Objective 3 Identify GNOME Desktop Components

The GNOME desktop includes one panel at the bottom of the screen.

**Figure 2-5**



There is a menu at the left side of the panel. This menu is labeled **Computer**. It is called the main menu.

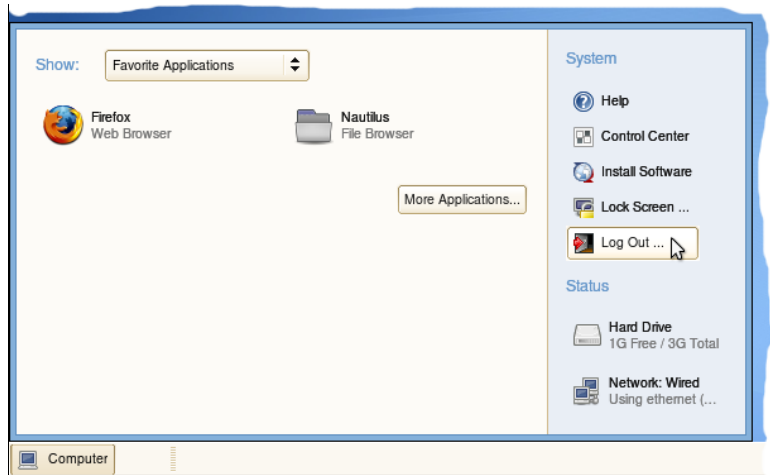
The empty space in the middle of the panel includes the task manager. All opened windows on the screen will be listed here.

At the right of the panel there are some more items. Which icons are available depends from your hardware:

- **Globe.** Searches for new updates.
- **Battery.** Power management for laptops.
- **Speaker.** Volume control.
- **Clock.** Shows date and time.
- **Board.** Minimizes all open windows or shows them again on the desktop.

You can start a programs with an icon on the desktop by double-clicking the icon. But normally programs are started from the main menu.

**Figure 2-6**

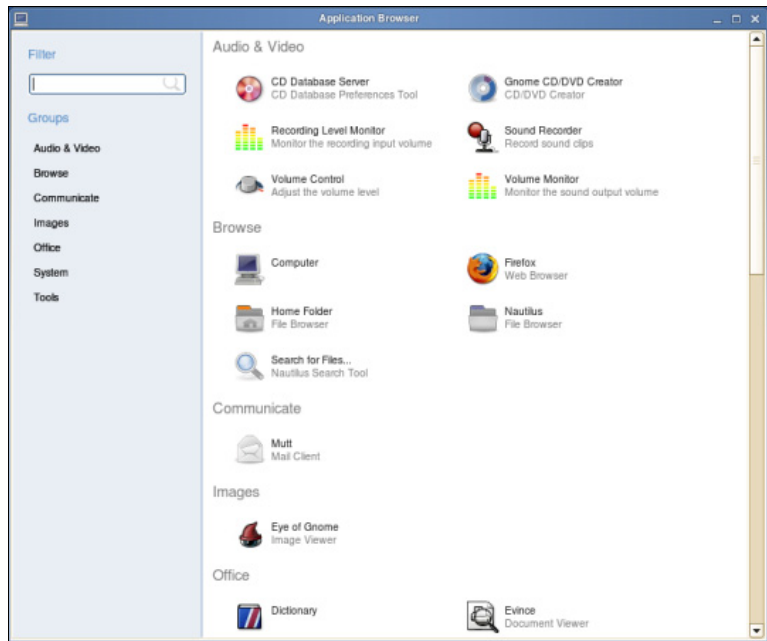


At the top of the left frame there is a pull-down menu showing three different filters:

- **Favorite Applications**
- **Recently Used Applications**
- **Recent Documents**

In the left frame, there is also a button labeled **More Applications**. If you select this button, the application browser appears.

**Figure 2-7**



The right frame of the application browser shows a list of the most important installed applications. The applications are grouped and you can see a list of the groups in the left frame. Select a group to see only the applications that belong to this group.

The filter option adds even more flexibility. Enter a part of the name of the application you want to start in the **Filter** textbox in the left frame. The filtered applications are shown immediately in the right frame.



In the right frame of the main menu, there are five system options:

- **Help.** Starts the online help.
- **Control Center.** Starts the GNOME Control Center where you can configure your desktop with.
- **Install Software.** Shows a list with the available software on your registered installation media.
- **Lock Screen.** Locks the screen. To unlock you have to enter your password.
- **Log Out.** Must be selected to log out of the system.

At the bottom of the right frame you can see the status of your hard drives and network.

To start an application select the icon in the main menu or the application browser with a single mouse click.

## Objective 4    Manage Icons in GNOME

You can manage icons on your desktop in different ways. For simplicity, we will describe only the most important methods.

You can find icons in the following three areas on your desktop:

- Desktop
- Panel
- Main Menu

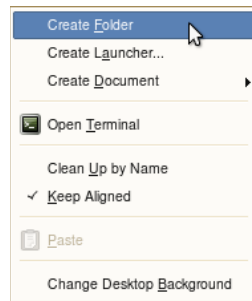
### ***Desktop***

To create an icon for an application on your desktop, select the item in your application menu, drag it to a free space on your desktop, and release the mouse button.

Notice there is a small plus icon at the mouse pointer when moving the icon. This indicates, that a copy of the icon will be created.

To create a new icon right-click a free space on your desktop. A menu pops up.

**Figure 2-8**



At the top of the pop-up menu, there are three entries to create a new item:

- **Create Folder.** Creates a new and empty folder icon.

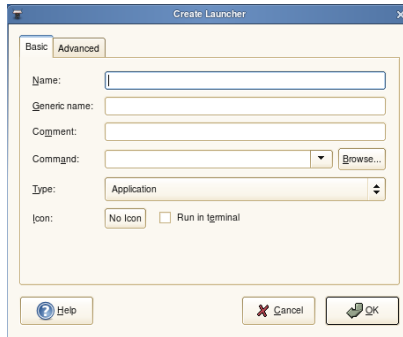
When the icon appears you can enter the folder's name.

**Figure 2-9**



- **Create Launcher.** Creates a new application launcher. A dialog appears:

**Figure 2-10**



Enter the following information:

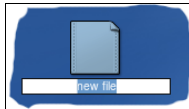
- **Name.** Name and label of the launcher.
- **Generic name.** (Optional) You can enter a generic name here.
- **Comment.** (Optional) This comment is shown as a tool tip when moving the mouse pointer over the icon.
- **Command.** Command that should be executed when double-clicking the launcher icon.
- **Type.** You can create launchers for different file types (e.g., application, directory, link, device) using this dialog.

- ❑ **Icon.** (Optional) Select an icon for the launcher.
- ❑ **Run in terminal.** Select this option if the application does not have a graphical user interface and runs in a terminal window.
- **Create Document.** You can create an empty document by using this menu.

Depending on your installed software there are various document types available in this menu. After a default installation there is only the possibility to create an empty text file.

When the icon appears you can enter the text file's name.

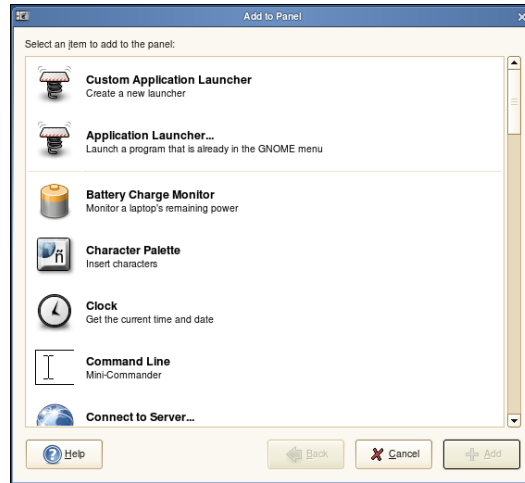
**Figure 2-11**



## Panel

You can add new programs to the bottom panel by right-clicking a free area of the panel and then selecting **Add to Panel**. From the dialog that appears, select the application you want to add.

**Figure 2-12**



You can remove a program from the control panel by right-clicking its icon in the bottom panel and then selecting **Remove From Panel**.

You can move icons in the panel by holding down the right mouse button and selecting **Move** from the Context menu.

## **Main Menu**

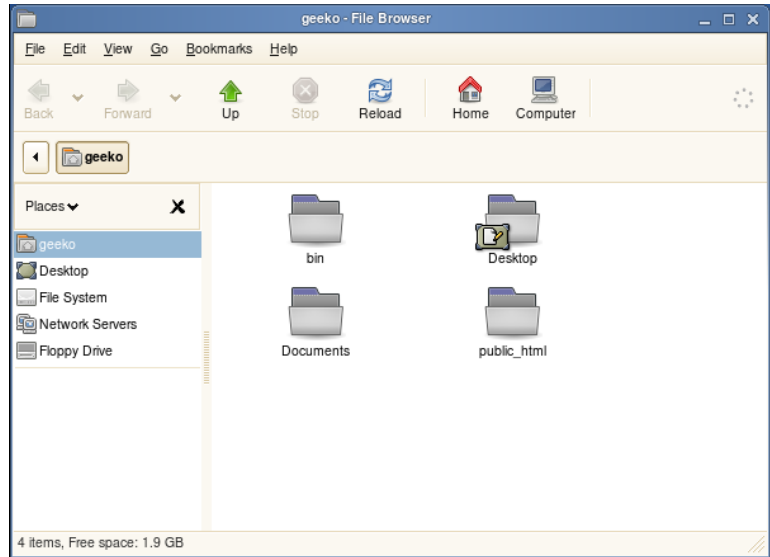
Only the user root is allowed to add a new entry to a menu. Normal users are only allowed to declare favorite applications. Therefore do the following:

1. Open the main menu in the panel.  
The menu appears.
2. Select **More Applications**.
3. Select an application item in the right frame with the right mouse button.
4. Select **Add to Favorites** from the pop-up menu.

## Objective 5 Use the GNOME File Manager (Nautilus)

GNOME provides its own file manager (called Nautilus):

Figure 2-13



You can start Nautilus by selecting the *username's* **Home** icon on the desktop or by selecting **Nautilus** from the main menu. By default Nautilus is marked as a favorite application.

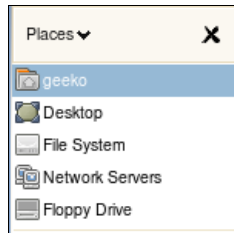
Normally Nautilus shows the content of the user's home directory after starting.

The left frame of the Nautilus windows shows the content of the current directory.

You can see your current position in the location bar below the tool bar. All higher directories are shown as buttons. Select one of these buttons to switch into the higher directory.

The right frame is called **Side Panel**.

**Figure 2-14**



At the top of the side panel there is a menu where you can select the content of the side panel:

- **Places.** Shows the most important directories and devices to store files.
- **Information.** Shows some information about the current directory.
- **Tree.** Shows the file system tree and the tree of the home directory.
- **History.** Shows a history of the last visited directories.
- **Notes.** Enter notes for the current directory.
- **Emblems.** Shows the list of emblems.

To add an emblem to an icon use drag and drop. **Erase** removes all emblems from an icon.

**Figure 2-15**





## Objective 6 Search for Files

Sometimes you need to find a file so you can edit it, but you do not know exactly where it is located in the file system. You might know the name of this file or only a part of the name.

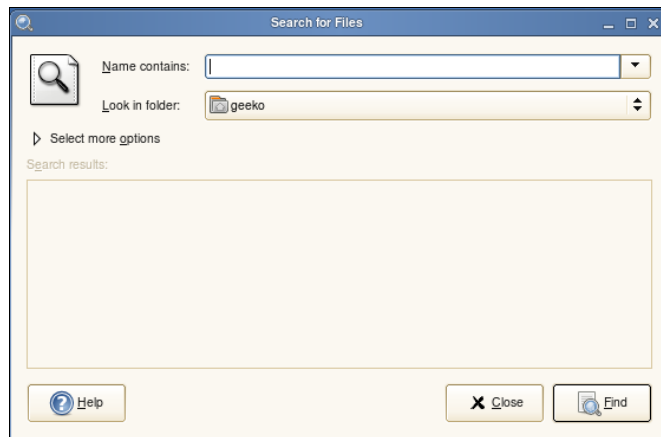
At another time, you might need a list of all files that have been modified in the last two days or that exceed a certain size.

If you enter **search** in the application browser, two applications are found:

- **Nautilus Search Tool (Browse application group).** The Nautilus file manager is used for searching files. This tool allows only to search for file names.
- **GNOME Search Tool (System application group).** This tool allows you to search for information such as file size, date, or file owner.

After selecting the GNOME Search tool from the application browser, the following dialog appears.

**Figure 2-16**



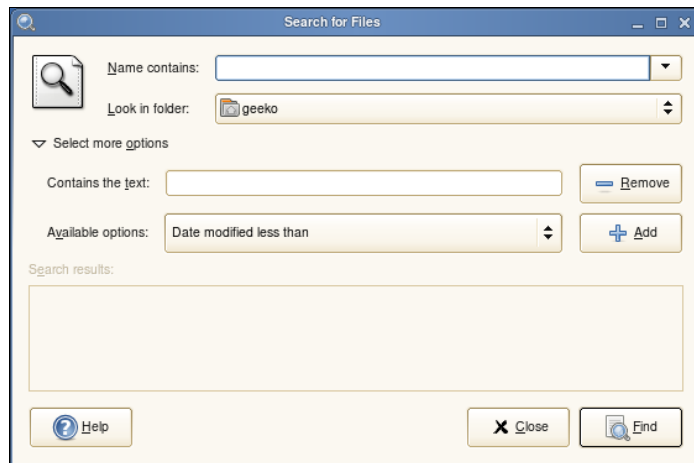
In the **Name contains** field, enter a part of the file name you want to find.

Enter the directory you want to search in **Look in folder**. Select **Find** to start the search process. All matching files and directories are shown in the lower window with details of their locations.

Further settings can be made when you open the menu under **Select more options**. Select a search rule from the pulldown menu **Available options**.

After selecting the **Add** button, a new text field is added and you can enter the information the option needs. To remove a search rule select the **Remove** button next to the rule.

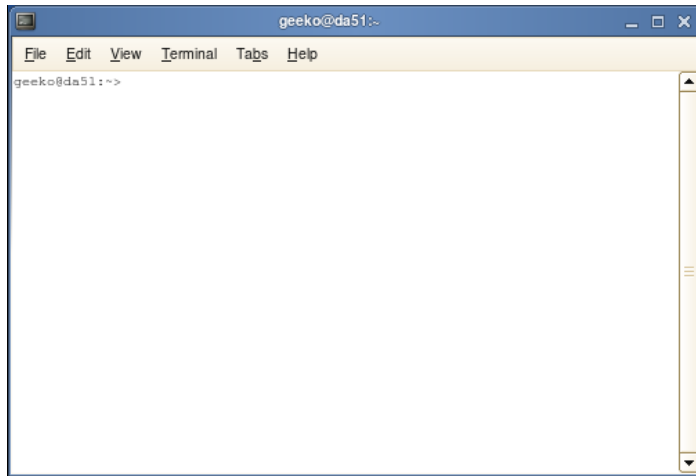
**Figure 2-17**



## Objective 7      Access the Command Line Interface From the Desktop

Besides using the virtual terminals, you can start a terminal emulation from your GNOME desktop by selecting **Gnome Terminal** (shown in the following picture) or **X Terminal** from the main menu. Both belong to the **System** application group.

**Figure 2-18**



The terminal opens inside a window with options you can select to modify the display of the terminal (such as font and background color).

## Summary

Objective	Summary
1. Log In	When the computer is booted and ready for work, the login dialog appears.
2. Log Out and Shut Down	When you are ready to log out of the system, open the <b>Computer</b> menu in the bottom panel.
3. Identify GNOME Desktop Components	<p>The menu labeled <b>Computer</b> is called the <i>main menu</i>.</p> <p>You can start a programs with an icon on the desktop by double-clicking the icon. But normally programs are started from the main menu.</p> <p>At the top of the left frame there is a pull-down menu showing three different filters.</p> <p>In the left frame, there is also a button labeled <b>More Applications</b>. If you select this button, the application browser appears.</p> <p>To start an application select the icon in the main menu or the application browser with a single mouse click.</p>
4. Manage Icons in GNOME	<p>You can find icons in the following three areas on your desktop:</p> <ul style="list-style-type: none"><li>■ Desktop</li><li>■ Panel</li><li>■ Main Menu</li></ul>

Objective	Summary
5. Use the GNOME File Manager (Nautilus)	<p>GNOME provides its own file manager (called Nautilus).</p> <p>Normally Nautilus shows the content of the user's home directory after starting.</p> <p>The right frame is called <b>Side Panel</b>.</p> <p>At the top of the side panel there is a menu where you can select the content of the side panel.</p>
6. Search for Files	<p>The <b>GNOME Search Tool</b> allows you to search for information such as file size, date, or file owner.</p> <p>Further settings can be made when you open the menu under <b>Select more options</b>. Select a search rule from the pulldown menu <b>Available options</b>.</p>
7. Access the Command Line Interface From the Desktop	<p>You can start a terminal emulation from your GNOME desktop by selecting Gnome Terminal (shown in the following picture) or X Terminal from the main menu.</p>



## SECTION 3    Manage Hardware

Although most hardware devices can be configured with YaST or are even automatically detected when plugged into the system, it is sometimes helpful to understand how things work in the background.

In this section, you are introduced to the SUSE Linux Enterprise Server 10 hardware management and how device drivers are loaded.

You also learn how to add and replace certain types of hardware.

### Objectives

1. Describe the Differences between Devices and Interfaces
2. Describe how Device Drivers Work
3. Describe how Device Drivers Are Loaded
4. Manage Kernel Modules Manually
5. Describe the sysfs File System
6. Describe how udev Works
7. Use the hwup Command

## Objective 1      **Describe the Differences between Devices and Interfaces**

This objective uses the terms “device” and “interface.” These terms are often confused, not only by users and administrators but also by developers of operating systems and related tools.

This course uses the following definitions for device and interface:

- **Device.** A device is a real, physical piece of hardware. This can be a PCI network card, an AGP graphic adapter, a USB printer, or any kind of hardware that you can hold, feel, or break if you want to.
- **Interface.** An interface is a software component associated with a device. To use a physical piece of hardware, it needs to be accessed by a software interface.

A device can have more than one interface.

Interfaces are usually created by a driver. In Linux, a driver is usually a software module that can be loaded into the Linux kernel. Therefore, a driver can be seen as the glue between a device and its interfaces.



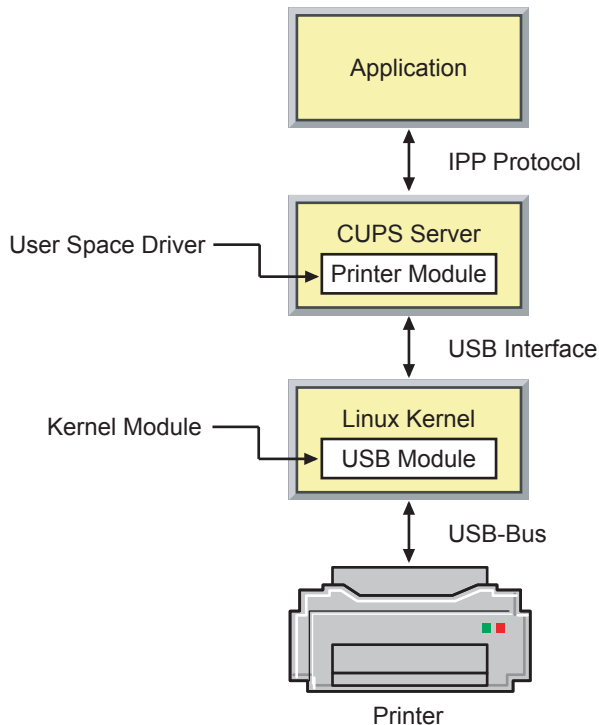
## Objective 2 Describe how Device Drivers Work

As described before, device drivers access and use a device. There are 2 basic kinds of device drivers:

- **Kernel modules.** The functionality of the Linux kernel can be extended by kernel modules. These modules can be loaded and removed during runtime. They allow the kernel to provide access to hardware.
- **User space drivers.** Some hardware needs additional drivers that work in user space. Examples of this kind of hardware are printers or scanners.

The following illustrates the roles of kernel and user space drivers:

**Figure 3-1**



While the handling of user space drivers depends on the framework they are used in, you can manage kernel modules with the following commands:

- **lsmod.** This command lists all loaded kernel modules. For example:

**lsmod**

- **modprobe.** This command loads kernel modules. Because kernel modules can depend on each other, modprobe automatically resolves these dependencies and loads all required modules. For example:

**modprobe usb-storage**

In this example, modprobe loads the usb-storage module which is needed to access storage devices connected with the USB bus.

Because this module requires other USB modules, modprobe also loads these modules.

- **rmmod.** This command removes loaded kernel modules. For example:

**rmmod usb-storage**

Only modules that are not needed can be removed. In this example, the USB device first has to be disconnected before the usb-storage module can be removed.

Kernel modules are files that are stored in the directory `/lib/modules/kernel-version/`.

Because modules normally work only with the kernel version they are built for, a new directory is created for every kernel update you install.

Modules are stored in several subdirectories with a filename extension of **.ko** for kernel object. When loading a module with `modprobe`, you can leave out the extension and use just the module name.

## Objective 3      Describe how Device Drivers Are Loaded

Because it would be very inconvenient to load all kernel modules manually after every system start, there are several methods to perform this task automatically.

The following is an overview of how device drivers are loaded in SUSE Linux Enterprise Server 10:

- **initrd.** Important device drivers that are necessary to access the root partition are loaded from initrd. initrd is a special file that is loaded into memory by the boot loader. Examples of such modules are the SCSI host controller and file system drivers.
- **initscripts.** Some initscripts are dedicated to loading and setting up hardware devices, such as the ALSA sound script for sound cards.
- **udev.** udev also loads kernel modules with the hwup command.
- **X Server.** Although the graphics card drivers are not kernel modules, X Server loads special drivers to enable hardware 3D support.
- **manually.** You can always load kernel modules on the command line or in scripts with the modprobe command or with the hwup or hwdown commands.

## Objective 4    **Manage Kernel Modules Manually**

Although SUSE Linux Enterprise Server 10 initializes most hardware devices automatically, it can be very helpful to know how kernel modules are managed manually.

To manage kernel modules, you need to understand the following:

- Kernel Module Basics
- Manage Modules from the Command Line
- modprobe Configuration File (/etc/modprobe.conf)



---

For the latest kernel documentation, see /usr/src/linux/Documentation.

---

### ***Kernel Module Basics***

The kernel that is installed in the directory /boot/ is configured for a wide range of hardware. It is not necessary to compile a custom kernel, unless you want to test experimental features and drivers.

Drivers and features of the Linux kernel can either be compiled into the kernel or be loaded as kernel modules. These modules can be loaded later, while the system is running, without having to reboot the computer.

This is especially true of kernel modules that are not required to boot the system. By loading them as components after the system boots, the kernel can be kept relatively small.

The kernel modules are located in the directory  
`/lib/modules/version/kernel/`.

For example, the modules for the 2.6 kernel can be found in the following directory:

**`/lib/modules/2.6.16-0.12-default/kernel/`**

## ***Manage Modules from the Command Line***

The following are commands you can use from a command line when working with modules:

- **lsmod.** This command lists the currently loaded modules in the kernel.

The following is an example:

```
DA50:~ # lsmod
Module                Size  Used by
quota_v2              12928  2
edd                  13720  0
joydev               14528  0
sg                   41632  0
st                   44956  0
sr_mod               21028  0
ide_cd               42628  0
cdrom                42780  2 sr_mod,ide_cd
nvram                13448  0
usbserial            35952  0
parport_pc           41024  1
lp                   15364  0
parport              44232  2 parport_pc,lp
ip_v6                276348  44
uhci_hcd              35728  0
intel_agp             22812  1
agpgart              36140  1 intel_agp
evdev                 13952  0
usbcore              116572  4 usbserial,uhci_hcd
```

The list includes information about the module name, size of the module, how often the module is used, and by which other modules use it.

- **insmod *module*.** This command loads the indicated *module* into the kernel.

The module must be stored in the directory `/lib/modules/version_number/`. However, it is recommended to use `modprobe` for loading modules.

- **rmmod *module*.** This command removes the indicated *module* from the kernel. However, it can only be removed if no processes are accessing hardware connected to it or corresponding services.

We recommend that you use **modprobe -r** for removing modules.

- **modprobe *module*.** This command loads the indicated *module* into the kernel or removes it (with option -r).

Dependencies of other modules are taken into account when using modprobe. In addition, modprobe reads in the file `/etc/modprobe.conf` for any configuration settings.

This command can only be used if the file `/lib/modules/version/modules.dep` created by the command `depmod` exists. This file is used to add or remove dependencies.

The kernel daemon (Kmod since kernel version 2.2.x) ensures that modules needed in the running operation are automatically loaded using modprobe (such as accessing the CD-ROM drive).



---

For more detailed information, enter **man modprobe**.

---

- **depmod.** This command creates the file `/lib/modules/version/modules.dep`. This file contains the dependencies of individual modules on each other.

When a module is loaded (such as with modprobe), `modules.dep` ensures that all modules it depends on are also loaded.

If the file `modules.dep` does not exist, it is created automatically when the system starts by the start script `/etc/init.d/boot`. For this reason, you do not need to create the file manually.

On SUSE Linux Enterprise Server 10, depmode also creates the file `modules.alias`, which is used by `hwup` and `modprobe` to determine which driver needs to be loaded for which device. Learn more about this file in the `hwup` objective in this section.

- **`modinfo option module`.** This command displays information (such as license, author, and description) about the module indicated on the command line.

The following is an example:

```
DA50:~ # modinfo isdn
license:      GPL
author:       Fritz Elfert
description:  ISDN4Linux: link layer
depends:       slhc
supported:    yes
vermagic:     2.6.5-7.21-default 586 REGPARM gcc-3.3
```



For more detailed information, enter **`man modinfo`**.

---



### ***modprobe Configuration File (/etc/modprobe.conf)***

The file `/etc/modprob.conf` is the configuration file for the kernel modules. For example, it contains parameters for the modules that access hardware directly.

The file plays an important role in loading modules. Various command types can be found in the file, such as the following:

- **install.** These instructions let modprobe execute commands when loading a specific module into the kernel.

The following is an example:

```
install      eth0      /bin/true
```

- **alias.** These instructions determine which kernel module will be loaded for a specific device file.

The following is an example:

```
alias      eth0      nvnet
```

- **options.** These instructions are options for loading a module.

The following is an example:

```
options      ne      io=0x300 irq=5
```



---

For more detailed information, enter **man 5 modprobe.conf**.

---

### **Exercise 3-1     Manage the Linux Kernel Modules**

In this exercise, you load and unload kernel modules.

Do the following:

1. From a terminal window, su to root (**su -**) with a password of **novell**.
2. View the currently loaded kernel modules by entering **lsmod**.
3. Scroll through the modules to see if the joystick module (**joydev**) is loaded.

The 0 in the Used column indicates that the module is not in use.

4. Remove the joystick module from the kernel memory by entering **rmmmod joydev**.
5. Verify that the joydev kernel module was removed from memory by entering **lsmod**.

Notice that the module joydev is no longer listed.

6. Load the joystick kernel module by entering **modprobe joydev**.
7. Verify that the joydev kernel module is loaded in memory by entering **lsmod**.
8. View the kernel modules configuration by entering the following:

**modprobe -c | less**

9. Scroll through the configuration information by pressing the **Spacebar**.
10. When you finish, return to the command line by typing **q**.
11. Create a list of kernel modules dependencies by entering **depmod -v | less**.

It takes a few moments for the information to be generated.

- 12.** Scroll through the dependency information by pressing the **Spacebar**.
- 13.** When you finish, return to the command line by typing **q**.
- 14.** Close the terminal window by entering **exit** twice.

***(End of Exercise)***

## Objective 5      Describe the sysfs File System

sysfs is a virtual file system that is mounted under /sys. In a virtual file system, there is no physical device that holds the information. Instead, the file system is generated virtually by the kernel.

The directory and file structure under /sys, provides information about the hardware which is currently connected with a system. Under /sys, there are 4 main directories:

- **/sys/bus** and **/sys/devices**. These directories contain different representations of system hardware. Devices are represented here.

For example, the following represents a digital camera connected to the USB bus:

```
/sys/bus/usb/devices/1-1/
```

This directory contains several files that provide information about the device. The following is a listing of the files in this directory:

1-1:1.0	bMaxPower	manufacturer
bcdDevice	bNumConfigurations	maxchild
bConfigurationValue	bNumInterfaces	power
bDeviceClass	detach_state	product
bDeviceProtocol	devnum	serial
bDeviceSubClass	idProduct	speed
bmAttributes	idVendor	version

For example, by reading the content from the manufacturer file, you can determine the manufacturer of the device:

```
cat manufacturer
OLYMPUS
```

In this case, an Olympus digital camera is connected with the system.

- **/sys/class** and **/sys/block**. The interfaces of the devices are represented under these 2 directories.

For example, the interface belonging to the Olympus digital camera is represented by the following directory:

**/sys/block/sda/**

The directory named /sda is the digital camera accessed like a SCSI hard disk.

The following is the content of the /sda directory:

```
dev      queue  removable  size
device   range  sda1       stat
```

The subdirectory /sda1 represents the interface to the first partition on the camera's memory card. For example, by reading the content of /sda1/size, you can determine the size of the partition:

```
cat sda1/size
31959
```

The partition has a size of 31959 512-byte blocks, which is about 16 MB.

To connect an interface with a device, file system links are used. In the Olympus digital camera example, a link exists from the file /sys/block/sda/device to the corresponding device:

```
ll device
lrwxrwxrwx 1 root root 0 Aug 17 14:03 device ->
../../../../devices/pci0000:00/0000:00:1d.0/usb1/1-1/1-1:1.0/hos
t0/0:0:0:0
```

In this way, all interfaces of the system are linked with their corresponding devices.

Beside the representation in **sysfs**, there are also the device files in the **/dev** directory.

These files are needed for applications to access the interfaces of a device. The name “device file” is a bit misleading, the name “interface file” would be more suitable.

## Objective 6      Describe how udev Works

Before you can use a hardware device, you need to load the appropriate driver module and set up the corresponding interface. For most devices in SUSE Linux Enterprise Server 10, this is done by udev.

To get an overview about udev, do the following:

- Understand the Purpose of udev
- Understand how udev Works
- Understand Persistent Interface Names

### ***Understand the Purpose of udev***

udev has three main purposes:

- **Create device files.** The most obvious task of udev is to create device files under **/dev** automatically when a device is connected to the system. Before, the **/dev** directory was populated with every device, that might appear in the system. This led to a very complex and confusing **/dev** directory, as most of the device files were actually not used.
- **Persistent device names.** Another advantage of udev is that it provides a mechanism for persistent device names.
- **Hotplug replacement.** In SUSE Linux Enterprise Server 10, udev also replaces the hotplug system, which was responsible for the initialization of hardware devices in previous versions. udev is now the central point for hardware initialization in SUSE Linux Enterprise Server 10.

## ***Understand how udev Works***

udev is implemented as a daemon (udevd), which is started at boot time through the script `/etc/init.d/boot.d/S03boot.udev`. udev communicates with the Linux kernel through the **uevent** interface. When the kernel sends out a uevent message that a device has been added or removed, udevd does the following, based on the udev rules:

- Initializes devices by calling `hwup`.
- Creates device files in `/dev`.
- Sets up network interfaces with `ifup`.
- Renames network interfaces, if necessary.
- Mounts storage devices which are identified as hotplug in `/etc/fstab`.
- Informs other applications through HAL (Hardware Abstraction Layer) about the new device.

To handle uevent messages, which have been issued before udevd was started, the udev start script triggers these missed events by parsing the `sysfs` filesystem. In previous SUSE Linux Enterprise Server versions, this part of the system initialisation was done by the `coldplug` script.

Everything udev does, depends on rules, defined in configuration files under `/etc/udev/rules.d/`, which are used to process a uevent.

A detailed description of udev rules is beyond the scope of this course. However, in the following you find the most important facts:

- udev rules are spread over several files, which are processed in alphabetical order. Each line in these files is a rule. Comments can be added with the `#` character.
- Each rule consists of multiple key value pairs. Example for a key value pair: `kernel=="hda"`



- There are two different key types:
  - **match keys.** These keys are used to determine, if rule should be used to process an event.
  - **assignment keys.** These keys determine what to do if a rule is processed.

There always has to be at least one match and one assignment key in rule.

- For every uevent, all rules are processed. Processing does not stop when a matching rule is found.

You can monitor the activity of udev with the tool **udevmonitor**. When you start the tool and change the hardware configuration (e.g. plug or unplug an USB device) the udev activities are displayed on the screen. For more details, you can start the tool the option **--env**.

### ***Understand Persistent Interface Names***

The interface files in the directory **/dev** are created and assigned to the corresponding hardware device when the device is recognized and initialized by a driver. Therefore the assignment between device and interface file depends on:

- The order in which device drivers are loaded.
- The order in which devices are connected to a computer.

This can lead to situations, where it's not clear which device file is assigned to which device. Consider the following example:

You have two USB devices, a digital camera and a flash card reader. Both devices are accessed as storage devices through the device files **/dev/sda**, **/dev/sdb** ...

Which device is assigned to which device file, usually depends on the order in which they are plugged in. The first devices gets sda and so on.

udev can help to make this more predictable. With the help of sysfs, udev can find out, which device is connected to which interface file. The easiest solution for persistent device names would be to rename the interface files, for example from `/dev/sda1` to `/dev/camera`.

Unfortunately, interface files can not be renamed under Linux. The only exception to this rule are network interfaces, which traditionally have no interface files under `/dev`.

Therefore udev uses a different approach. Instead of renaming an interface file, a link with a unique and persistent name is created to the assigned interface file.

By default udev is for example configured to create these links for all storage devices. For each device, a link is created in each of the following subdirectories under **`/dev/disk/`**:

- **by-id**. The name of the link is based on the vendor and on the name of a device.
- **by-path**. The name of the link is based on the bus position of a device.
- **by-uuid**. The name of the link is based on the serial number of a device.

The udev rules which create these links are located in the file `/etc/udev/rules.d/60-persistent-storage.rules`.

This means, that the association between devices and interface files still depends on the order in which the drivers are loaded or in which order devices are connected with the system.

With udev however, persistent links are created and adjusted everytime the device configuration changes.

As mentioned before, network interfaces are treated differently. They don't have interface files and they can be directly renamed by udev.

Persistent network interfaces are configured as udev rule in the file **/etc/udev/rules.d/30-net\_persistent\_names.rules**

The following is an example rule:

```
SUBSYSTEM=="net", ACTION=="add", SYSFS{address}=="00:30:05:4b:98:85",  
IMPORT="/sbin/rename_netiface %k eth0"
```

The matching key `SYSFS{address}` is used to identify a network device by its MAC address. At the end of the rule, the name of the interface is given. In this example `eth0`.



In SUSE Linux Enterprise Server 9, it was possible to configure persistent network interface names in the interface configuration files under `/etc/sysconfig/network`. This is not supported anymore in SUSE Linux Enterprise Server 10. Interface names now have to be configured in the udev rule.

---

### **Exercise 3-2     Add a device symlink with udev**

In this exercise, you create a udev rule, that creates a symlink in `/dev` when a device is plugged in. To perform this exercise, you need a USB mouse.

Do the following:

1. Open a terminal window and **su-** to the root user.
2. **cd** to the `/dev` directory.
3. Make sure that there is no symlink or device **geekomouse** in the `/dev` directory. This can be done with the command:  
**ls geekomouse.**
4. Open the file `/etc/udev/rules.d/60-persistent-input.rules` with a text editor.
5. Identify the two rules, which are introduced with the **#by-id links** comment.

The rule with the matching key **KERNEL=="mouse\*"** creates symlinks of each mouse device under `/dev/input/by-id/`

The names of these symlinks are generated from hardware parameters like the serial number of the mouse. This way a persistent and unique device name is created.

6. Duplicate the mouse line to create a new rule.
7. In the new rule, change the value of the SYMLINK key to **geekomouse** (**SYMLINK+="geekomouse"**).
8. Save and close the file.
9. Unplug your USB mouse, wait a few seconds and plug it in again.
10. Check again, if the symlink `/dev/geekomouse` exists.
11. Unplug the mouse again, and see how the symlink is automatically removed by udev.

**(End of Exercise)**

## Objective 7    Use the hwup Command

The command **hwup** is used by **udev** to load driver modules and to initialize devices. In this objective, you learn how to use this command and how to interpret the corresponding configuration files.

To start a device, **hwup** is called with a hardware description as argument. The hardware description is a unique identifier for a specific device in the system. The following is an example **hwup** command line:

**hwup bus-pci-0000:02:08.0**

The device description consists of the following components:

- **bus.** This determines that the device is identified by the bus it is connected to.
- **pci.** This indicates that the device is connected to the PCI bus.
- **0000:02:08.0.** This is the address of the device in the PCI bus.

You can display the PCI address of a device with the **lspci** command, as in the following example:

```
...
0000:02:08.0 Ethernet controller: Intel Corp. 82801BD
PRO/100 VE (LOM) Ethernet Controller (rev 81)
...
```

As you can see, with the command **hwup bus-pci-0000:02:08.0**, the ethernet controller with the PCI address 0000:02:08.0 would be started.

The command **hwdown <hardware\_description>** can be used to stop a device. **hwdown** is a link to **hwup** and not a separate command.

hwdown unbinds a device from it's driver, but does not unload the driver module automatically.

There are two different ways how hwup gets information about devices and about the driver that needs to be loaded for a device:

- From Configuration Files
- From sysfs

### ***From Configuration Files***

When a device needs to be initialized, hwup first tries to read a device configuration from files in the directory **/etc/sysconfig/hardware/**.

In order to determine the correct configuration file, the configuration filenames follow a specific naming scheme.

The following is the filename for a PCI network adapter:

**hwcfg-bus-pci-0000:02:08.0**

The filename consists of the keyword **hwcfg** and the hardware description of a device.

The following lists the possible variables in a device configuration file:

Table 3-1	Variable	Description
	STARTMODE	<p>This determines when and how a device will be started:</p> <ul style="list-style-type: none"><li>■ <b>auto.</b> The device is automatically started at boot time or by udev when the device is connected to the system.</li><li>■ <b>manual.</b> The device <i>should not</i> be started automatically, but it <i>can</i> be started manually.</li><li>■ <b>off.</b> The device should never be started.</li></ul>
	MODULE	<p>The value of this variable determines the name of the kernel module that should be loaded for the device.</p> <p>If multiple modules have to be loaded, you can use this variable multiple times with any suffix appended.</p> <p>You must then use the same suffixes for multiple MODULE_OPTIONS variables.</p> <p>Example:</p> <pre>MODULE_A="foo" MODULE_B="bar" MODULE_OPTIONS_A="foo-opt" MODULE_OPTIONS_B="bar-opt1=xyz"</pre>
	MODULE_OPTIONS	<p>With this variable, options can be passed to the kernel module.</p>

(continued) **Table 3-1**

Variable	Description
SCRIPT{UP,DOWN}_[type]	This specifies the script to be called for initialization and deconfiguration of a specific device type.  This script is called if the type of the device to be initialized matches the type given in this parameter.
SCRIPT{UP,DOWN}	This specifies the script to be called for initialization and deconfiguration of the device.  It will be called only if no matching type-specific scripts are configured.

The following is an example of a configuration file for a network adapter:

```
MODULE='e100'  
MODULE_OPTIONS=' '  
STARTMODE='auto'
```

The module e100 is loaded, there are no options for this modul and the device is started automatically at boot time.

The hwup command is usually called by udev, but you can also use it manually. For example, the following command starts the network card shown in the previous:

**hwup bus-pci-0000:02:08.0**

The last 3 elements of the configuration filename specify the device.

You can use the command hwdown to deconfigure devices, as in the following:

**hwdown bus-pci-0000:02:08.0**



## ***From sysfs***

When there is no configuration file for a device under `/etc/hardware`, `hwup` uses **sysfs** to find out more about the required driver.

In the file `/sys/bus/devices/<pci_device_id>/modalias` a unique id for a device can be found. The following is the content of the file `modalias` for a network adapter:

```
pci:v00008086d00001039sv00001734sd00001001bc02sc00i00
```

Now `udev` calls the command `modprobe` and uses the `modalias` as parameter.

The file `/lib/modules/<kernel_version>/modules.alias` contains a list with `modalias` IDs and their corresponding kernel driver modules. The information about which driver module handles which `modalias` ID is included in the source files of the modules. The command `depmod` is used to extract the information from the modules and to write them into the file `modules.alias`.

Therefore you can consider the file **modules.alias** as a kind of driver database, which is used to find out, which device is handled by which driver.

When called from `udev` with the `modalias` parameter, `modprobe` looks for the driver in `modules.alias` and load the associated driver.

To prevent `udev/hwup/modprobe` from loading a driver automatically in the described way, a driver can be entered in the file `/etc/modprobe.d/blacklist`. By default the ALSA sound drivers are for example entered here, because these drivers are loaded by the ALSA init script (`alsasound`).

### **Exercise 3-3      Explore Hardware Initialization**

In this exercise, you learn how to shutdown a device with `hwdown` and how to start it again manually. The exercise consists of the following parts:

- Part I: Stop the Ethernet Adapter with `hwdown`
- Part II: Unload the Driver Module
- Part III: Load the Driver Module with a modalias

#### **Part I: Stop the Ethernet Adapter with `hwdown`**

Do the following:

1. Open a terminal window and **su-** to the root user.
2. Enter the command **ip a**. You should see a list with your network interfaces. (At least `eth0`).
3. Enter the command **lspci**.
4. Look for a device with the description **Ethernet controller** and note the PCI ID of this device bellow this step. If you have more than one ethernet adapter, choose the first one

- 
5. Enter the command:  
**`hwdown bus-pci-0000:<network_adapter_pci_id>`**
  6. Enter the command **ip a** again. The interface of the device which has been shutdown with `hwdown` should not be visible anymore.

#### **Part II: Unload the Driver Module**

1. Change into the directory **`/etc/sysconfig/hardware/`**.

2. Open the configuration file of the ethernet adapter with a text editor:

**hwcfg-bus-pci-0000:<network\_adapter\_pci\_id>**

3. From the **MODULE** option in the configuration file, note the name of the kernel driver module below this step .

-----

4. Enter the command **lsmod | grep <module\_name>**. As you can see, is the module still loaded.
5. Unload the module with the command **rmmod <module\_name>**
6. Verify with **lsmod | grep <module\_name>**, that the module is not loaded anymore.

### **Part III: Load the Driver Module with a modalias**

1. Change into the directory **/sys/bus/pci/devices/**.
2. Change into the directory **0000:<network\_adapter\_pci\_id>**
3. Enter **cat modalias** to display the modalias of the ethernet adapter.
4. Enter the command **modprobe <modalias>**. Use copy and paste from the cat output to enter the modalias.
5. Verify with **lsmod | grep <module\_name>**, that modprobe has detected and loaded the driver for the network adapter.
6. Enter the command **ip a**, and verify that the network interface is available again.
7. Reboot your system, to restore all network settings.

***(End of Exercise)***

## Summary

Objective	Summary
1. Describe the Differences between Devices and Interfaces	<p>The terms device and interface are often confused. This section uses the following definitions:</p> <ul style="list-style-type: none"><li>■ <b>Device.</b> A device is a physical piece of hardware.</li><li>■ <b>Interface.</b> An interface is a software component that is used to access a device.</li></ul> <p>One device can have more than one interface.</p> <p>An interface is created by a device driver.</p>
2. Describe how Device Drivers Work	<p>There are 2 basic kinds of device drivers:</p> <ul style="list-style-type: none"><li>■ <b>Kernel modules.</b> Kernel modules are loaded into the Linux kernel and extend its functionality.</li><li>■ <b>User space drivers.</b> These drivers run within user space applications.</li></ul> <p>Some devices require both: kernel modules and user space drivers.</p> <p>You can use the following commands to manage kernel modules:</p> <ul style="list-style-type: none"><li>■ <b>lsmod.</b> Use lsmod to list loaded drivers.</li><li>■ <b>modprobe.</b> Use modprobe load kernel modules.</li></ul>

Objective	Summary
2. Describe how Device Drivers Work (continued)	<ul style="list-style-type: none"><li>■ <b>rmmod.</b> Use rmmod to remove loaded kernel modules.</li></ul> <p>The kernel modules are files that are stored in the directory <code>/lib/modules/kernel-version/</code>.</p>
3. Describe how Device Drivers Are Loaded	<p>In a SUSE Linux Enterprise Server 10 system, kernel modules are loaded in the following ways:</p> <ul style="list-style-type: none"><li>■ From initrd</li><li>■ By initscripts</li><li>■ By udev</li><li>■ By the X Server</li><li>■ Manually by the user root</li></ul>
4. Manage Kernel Modules Manually	<p>The <code>hwup</code> command is used to start preconfigured devices.</p> <p>The device configuration files are stored in the directory <code>/etc/sysconfig/hardware/</code>.</p> <p>The filename of the configuration file contains a unique identifier for the corresponding device.</p> <p>In the configuration file, the following variables can be used:</p> <ul style="list-style-type: none"><li>■ <code>STARTMODE</code></li><li>■ <code>MODULE</code></li><li>■ <code>MODULE_OPTIONS</code></li><li>■ <code>SCRIPT{UP,DOWN}_[type]</code></li><li>■ <code>SCRIPT{UP,DOWN}</code></li></ul>

Objective	Summary
5. Describe the sysfs File System	<p>The sysfs file system provides a representation of all devices and interfaces of a system.</p> <p>Devices are represented in the directories: <code>/sys/bus</code> and <code>/sys/devices</code>.</p> <p>Interfaces are represented by the directories <code>/sys/class</code> and <code>/sys/block</code>.</p> <p>A device and its interfaces are connected with file system links.</p>
6. Describe how udev Works	<p>In SUSE Linux Enterprise Server 10, udev also replaces the hotplug system, which was responsible for the initialization of hardware devices in previous versions. udev is the central point for hardware initialization in SUSE Linux Enterprise Server 10.</p>
7. Use the hwup Command	<p>hwup is used to load device drivers. Information about the required drivers for a device is either taken from a configuration file under <code>/etc/sysconfig/hardware/</code> or from the file <code>/lib/modules/kernel_version/modules.alias</code>.</p>

## SECTION 4    Configure Linux File System Partitions

This section introduces some command you need to manage your Linux file system partitions.

### Objectives

1. Finalize Partitioning
2. Configure LVM with Command Line Tools

## Objective 1      Finalize Partitioning

The program **fdisk** is used for partitioning hard disks from the command line.

```
da10:~ # fdisk /dev/sda
```

```
The number of cylinders for this disk is set to 1111.  
There is nothing wrong with that, but this is larger than 1024,  
and could in certain setups cause problems with:  
1) software that runs at boot time (e.g., old versions of LILO)  
2) booting and partitioning software from other OSs  
   (e.g., DOS FDISK, OS/2 FDISK)
```

```
Command (m for help):
```

To actually write your changes to the partition table on the disk, enter **w** (write).

```
Command (m for help): w
```

```
The partition table has been altered!
```

```
Calling ioctl() to re-read partition table.
```

```
WARNING: Re-reading the partition table failed with error 16: Device or  
resource busy.
```

```
The kernel still uses the old table.
```

```
The new table will be used at the next reboot.
```

```
Syncing disks.
```



---

When the new table is written, you are not asked for confirmation if you really want to do this.

---

As the output of **fdisk** says, you cannot directly use the new partition to create a file system on a new partition. You could now reboot as suggested, but you can also use the program **partprobe** to get the kernel to use the new partition table.



## Objective 2      **Configure LVM with Command Line Tools**

Setting up LVM consists of several steps, with a dedicated tool for each:

- Tools to Administer Physical Volumes
- Tools to Administer Volume Groups
- Tools to Administer Logical Volumes

This is just a brief overview, not all available LVM tools are covered. To view the tools that come with LVM, enter **rpm -ql lvm2 | less** on a command line, and have a look at the corresponding manual pages for details on each of them.

### ***Tools to Administer Physical Volumes***

Partitions or entire disks can serve as physical volumes for LVM.

The ID of a partition used as part of LVM should be **Linux LVM, 0x8e**. However the ID **0x83, Linux**, works as well.

To use an entire disk as physical volume, it may not contain a partition table. Overwrite any existing partition table with **dd**:

```
da10:~ # dd if=/dev/zero of=/dev/hdd bs=512 count=1
```

The next step is to initialize the partition for LVM. The tool to use is **pvcreate**:

```
da10:~ # pvcreate /dev/hda9
Physical volume "/dev/hda9" successfully created
```

**pvscan** shows the physical volumes and their use:

```
da10:~ # pvscan
PV /dev/hda9    lvm2 [242,95 MB]
Total: 1 [242,95 MB] / in use: 0 [0    ] / in no VG: 1 [242,95 MB]
```

The tool **pvmove** is used to move data from one physical volume to another (providing there is enough space), in order to remove a physical volume from LVM.

### ***Tools to Administer Volume Groups***

The tool **vgcreate** is used to create a new volume group. To create the volume group system, and add the physical volume /dev/hda9 to it, enter:

```
da10:~ # vgcreate system /dev/hda9
Volume group "system" successfully created
da10:~ # pvscan
PV /dev/hda9    VG system    lvm2 [240,00 MB / 240,00 MB free]
Total: 1 [240,00 MB] / in use: 1 [240,00 MB] / in no VG: 0 [0    ]
```

pvscan shows the new situation.

To add further physical volumes to the group, use **vgexpand**. Removing unused physical volumes is done with **vgreduce** after shifting data from the physical volume scheduled for removal to other physical volumes using **pvmove**. **vgremove** removes a volume group, providing there are no logical volumes in the group.

## ***Tools to Administer Logical Volumes***

To create a logical volume, use **lvcreate**, specifying the size, the name for the logical volume, and the volume group:

```
da10:~ # lvcreate -L 100M -n data system
Logical volume "data" created
```

The next step is to create a file system within the logical volume and mount it:

```
da10:~ # lvscan
ACTIVE                '/dev/system/data' [100,00 MB] inherit
da10:~ # mkreiserfs /dev/system/data
mkreiserfs 3.6.19 (2003 www.namesys.com)
...
ReiserFS is successfully created on /dev/system/data.
da10:~ # mount /dev/system/data /data
```

As shown above, **lvscan** is used to view the logical volumes. It shows the device to use for the formatting and mounting.

**lvextend** is used to increase the size of a logical volume. After that you can increase the size of the file system on that logical volume to make use of the additional space.

Before you use **lvreduce** to reduce the size of a logical volume, you have to reduce the size of the file system. Only then reduce the size of the logical volume. If you cut off parts of the file system by simply reducing the size of the logical volume without shrinking the file system first, you will loose data.

## Summary

Objective	Summary
1. Finalize Partitioning	After changing the partitions using <code>fdisk</code> , you cannot directly create a file system on the new partition. You could now reboot, but you can also use the program <b><code>partprobe</code></b> .

Objective	Summary
2. Configure LVM with Command Line Tools	<p>To initialize a partition for LVM, use the tool <b>pvcreate</b>.</p> <p><b>pvscan</b> shows the physical volumes and their use.</p> <p>The tool <b>pvmove</b> is used to move data from one physical volume to another.</p> <p>The tool <b>vgcreate</b> is used to create a new volume group.</p> <p>To add further physical volumes to the group, use <b>vgexpand</b>.</p> <p>Removing unused physical volumes is done with <b>vgreduce</b> after shifting data from the physical volume scheduled for removal to other physical volumes using <b>pvmove</b>.</p> <p><b>vgremove</b> removes a volume group, providing there are no logical volumes in the group.</p> <p>To create a logical volume, use <b>lvcreate</b>.</p> <p><b>lvscan</b> is used to view the logical volumes.</p> <p><b>lvextend</b> is used to increase the size of a logical volume.</p> <p>Before you use <b>lvreduce</b> to reduce the size of a logical volume, you have to reduce the size of the file system.</p>



## **SECTION 5**    Use the NetworkManager to Configure the Network

In case you are using SUSE Linux Enterprise Server 10 on a laptop, you will most likely use different kinds of Internet access, depending on where you are—maybe a LAN in your office and a wireless connection at a customer site.

### **Objective**

1. Use the NetworkManager to Configure the Network

## Objective 1      **Use the NetworkManager to Configure the Network**

The conventional network setup requires you to switch to the root account to change the network configuration. The purpose of the **NetworkManager** (package `NetworkManager`) is to allow the user to change the network configuration according to his needs, without switching to the root account.

NetworkManager runs as a root-user system level daemon, since root privileges are needed to manipulate hardware directly. The programs used for this purpose are `/usr/sbin/NetworkManager` and `/usr/sbin/NetworkManagerDispatcher`. **nm-tools** can be used to list information about NetworkManager, devices, and wireless networks.

From a list of all adapters currently installed on the system, NetworkManager will first try a wired and then a wireless adapter. Wireless adapters that support wireless scanning are preferred over ones that cannot. NetworkManager does not try to keep a connection up as long as possible, meaning that plugging into a wired network will switch the connection to the wired network, away from the wireless one.

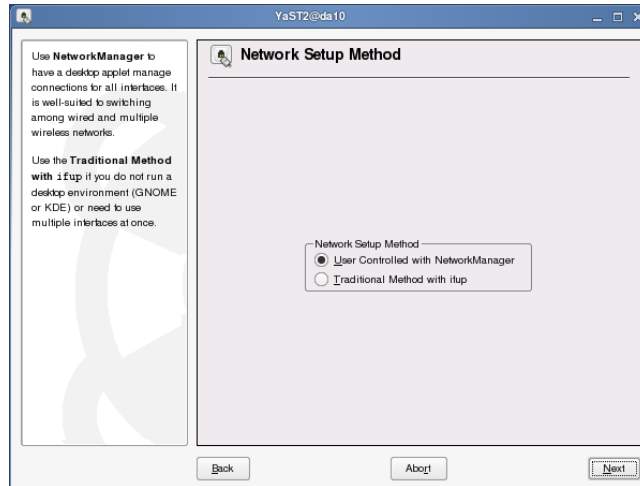
For wireless networking support, NetworkManager keeps two lists of wireless networks: a **Trusted list**, and a **Preferred list**. The trusted list contains networks the user specifically adds to it, while the preferred list contains networks the user forces NetworkManager to connect to.

Since trusted and preferred networks are user-specific, there must be some mechanism of getting and storing this information per user. This is achieved with a desktop-level per-user process, **nm-applet**, or `KNetworkManager` in KDE. NetworkManager communicates over DBUS with these user level processes.



Switching to NetworkManager is done by starting YaST and selecting **Network Devices > Network Cards**. In the **Network Setup Method** dialog, you select **User Controlled with NetworkManager**:

**Figure 5-1**

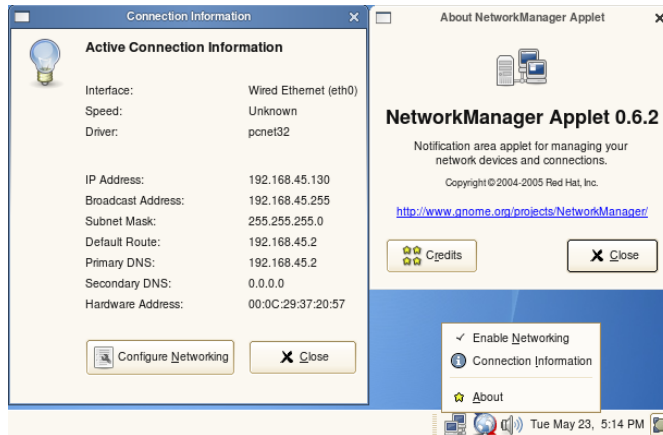


The following dialogs of this module are the same for both setup methods.

When selecting **User Controlled with NetworkManager**, YaST sets the variable **NETWORKMANAGER=** in `/etc/sysconfig/network/config` to “yes”.

Choosing the NetworkManager in YaST will also automatically start the Network Applet when a user logs in. Using the desktop applet, the user can easily change the network configuration:

**Figure 5-2**



Note: As there was no wireless card built into the computer on which the above screenshot was taken, there is no option for switching networks in this screenshot.

## Summary

Objective	Summary
1. Use the NetworkManager to Configure the Network	<p>NetworkManager allows the user to change the network configuration without having to assume root privileges.</p> <p>NetworkManager is mainly useful for use on laptops.</p>

---



## SECTION 6 Administer User Access and Security

In this section you learn how to perform basic user and group management tasks that provide users with a secure and accessible SUSE Linux Enterprise Server environment.

### Objectives

1. Configure User Authentication with PAM
2. Configure Security Settings

## Objective 1      **Configure User Authentication with PAM**

User authentication plays a central role in IT security. Linux uses PAM (Pluggable Authentication Modules) in the authentication process as a layer between users and applications. A Linux system administrator can use these modules to configure the way programs should authenticate users.

By providing system-wide access to applications through authentication modules, authentication does not have to be part of each application requiring authentication. The Pluggable Authentication Modules take care of that task for applications.

For example, when a user logs into a Linux system on a virtual terminal, a program called **login** is usually involved in this process.

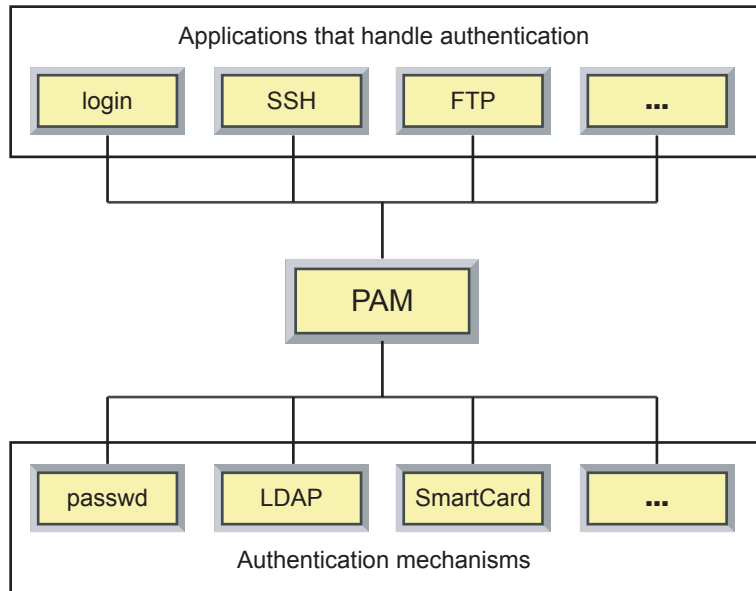
Login requires a user's login name and the password. The password is encrypted and then compared with the encrypted password stored in an authentication database. If the encrypted passwords are identical, login grants the user access to the system by starting the user's login shell.

If other authentication procedures are used, such as smart cards, all programs that perform user authentication must be able to work together with these smart cards. Before PAM was introduced all applications that handle authentication, like login, FTP, or SSH, had to be extended to support a smart card reader.

PAM makes things easier. PAM creates a software level with clearly defined interfaces between applications (such as login) and the current authentication mechanism. Instead of modifying every program, a new PAM module can to enable authentication with a smart card reader. After adjusting the PAM configuration for an application this application can make use of this new authentication method.

The following graphic illustrates the role of PAM:

**Figure 6-1**



Third party vendors can supply other PAM modules to enable specific authentication features for their products, such as the PAM modules that enable Novell's Linux User Management (LUM) authentication with eDirectory.

To understand how to configure PAM, you need to know the following:

- Location and Purpose of PAM Configuration Files
- PAM Configuration
- PAM Configuration File Examples
- Secure Password Guidelines
- PAM Documentation Resources

## ***Location and Purpose of PAM Configuration Files***

PAM provides a variety of modules—each one with a different purpose. For example, one module checks the password, another verifies the location from which the system is accessed, and another reads user-specific settings.

Every program that relies on the PAM modules has its own configuration file **/etc/pam.d/program\_name**. For example, the configuration file for the program `passwd` is called `/etc/pam.d/passwd`.

There is one special configuration file with the name **other**. This file contains the default configuration if no application-specific file is found.

In addition, there are global configuration files for most PAM modules in `/etc/security/`, which define the exact behavior of these modules. These include files such as `pam_env.conf`, `pam_pwcheck.conf`, `pam_unix2.conf`, and `time.conf`.

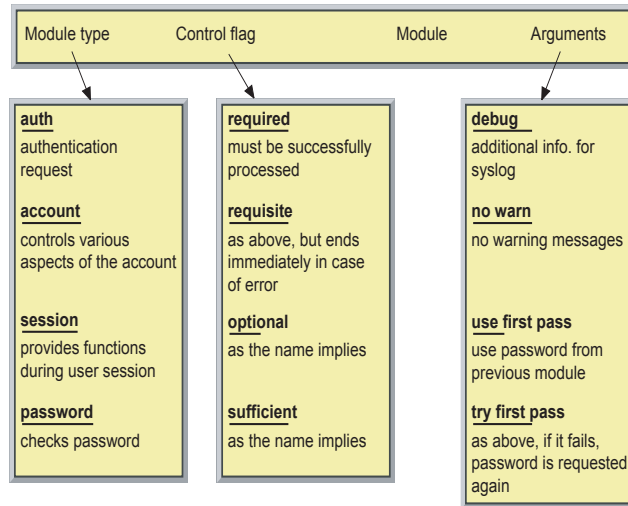
Every application that uses a PAM module actually calls a set of PAM functions. These functions are implemented in modules which perform the authentication process according to the information in the various configuration files and return the result to the calling application.



## PAM Configuration

Each line in a PAM configuration file contains 3 columns plus optional arguments:

**Figure 6-2**



The following describes the purpose of each column:

- **Module Type.** There are four types of PAM modules:
  - **auth.** These modules provide two ways of authenticating the user.  
  
First, they establish that the user is who he claims to be by instructing the application to prompt the user for a password or other means of identification.  
  
Second, the module can grant group membership or other privileges through its credential granting properties.
  - **account.** These modules perform nonauthentication based account management.

They are typically used to restrict or permit access to a service based on the time of day, currently available system resources (maximum number of users) or perhaps the location of the applicant user (for example, to limit 'root' login to the console).

- **session.** These modules are associated with performing tasks that need to be done for the user before she can be given access to a service or after a service is provided to her.

Such things include logging information concerning the user, mounting directories and the opening and closing of some data exchange with another user.

- **password.** This last module type is required for updating the authentication token associated with the user. Typically, there is one module for each challenge/response-based authentication (auth) module type.

- **Control Flag.** The control flag indicates how PAM will react to the success or failure of the module it is associated with.

Since modules can be stacked (modules of the same type execute in a series, one after another), the control-flags determine the relative importance of each module.

The Linux-PAM library interprets these keywords in the following manner:

- **required.** A module with this flag must be successfully processed before the authentication can proceed.

After the failure of a module with the required flag, all other modules with the same flag are processed before the user receives a message about the failure of the authentication attempt. This prevents the user from knowing at what stage their authentication failed.

- **requisite.** A module with this flag must also be processed successfully. In case of success, other modules are subsequently processed, just like modules with the required flag.

However, in case of failure the module gives immediate feedback to the user and no further modules are processed.

You can use the requisite flag as a basic filter, checking for the existence of certain conditions that are essential for a correct authentication.

- **optional.** The failure or success of a module with this flag does not have any direct consequences.

You can use this flag for modules that are only intended to display a message (such as telling a user that mail has arrived) without taking any further action.

- **sufficient.** After a module with this flag has been successfully processed, the calling application receives an immediate message about the success and no further modules are processed (provided there was no preceding failure of a module with the “required” flag).

The failure of a module with the sufficient flag has no direct consequences. In other words, any subsequent modules are processed in their respective order.

- **include.** This is not really a control flag but indicates that the keyword in the next column is to be interpreted as a file name relative to `/etc/pam.d/` that should be included at this point.

The file included has to have the same structure as any other PAM configuration file.

The purpose of **include** files is to simplify changes concerning several applications.

- **Module.** The PAM modules are located in the directory `/lib/security/`. Every filename of a module starts with the prefix `pam_`. You do not need to include the path, as long as the module is located in the default directory `/lib/security/`.



---

For all 64 bit platforms supported by SUSE Linux, the default directory is `/lib64/security/`.

---

Some PAM modules (such as `pam_unix2.so`) can be used for several module types (for instance type `auth` as well as type `password`).

- **Arguments (options).** You can include options in this column for the module, such as **debug** (enables debugging) or **nullok** (allows the use of empty passwords).

### ***PAM Configuration File Examples***

The following is the default configuration file for the login program on SLES 10, `/etc/pam.d/login`:

```
#%PAM-1.0
auth      required      pam_securetty.so
auth      include       common-auth
auth      required      pam_nologin.so
account   include       common-account
password  include       common-password
session   include       common-session
session   required      pam_lastlog.so nowtmp
session   required      pam_resmgr.so
session   optional      pam_mail.so standard
```

As an example of the files included in the above configuration, the file `/etc/pam.d/common-auth` looks like this::

```
#
# /etc/pam.d/common-auth - authentication settings common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authentication modules that define
# the central authentication scheme for use on the system
# (e.g., /etc/shadow, LDAP, Kerberos, etc.). The default is to use the
# traditional Unix authentication mechanisms.
#
auth      required      pam_env.so
auth      required      pam_unix2.so
```

The modules perform the following tasks (not all are included in the above configuration):

- **auth required pam\_securetty.so**

This module checks the file `/etc/securetty` for a list of valid login terminals. If a terminal is not listed in that file, the login is denied from that terminal. This concerns only the root user.

- **auth required pam\_env.so**

This module can be used to set additional environment variables. The variables can be configured in the file `/etc/security/pam_env.conf`.

- **auth required pam\_unix2.so nullok**

The module `pam_unix2.so` is used during the authentication process to validate the login and password provided by the user.

- **auth required pam\_nologin.so**

This module checks whether a file `/etc/nologin` exists. If such a file is found, its content is displayed when a user tries to log in. Login is denied for all but the root user.

- **account required pam\_unix2.so**

In this entry the `pam_unix2.so` module is used again, but in this case it checks whether the password of the user is still valid or if the user needs to create a new one.

- **password required pam\_pwcheck.so nullok**

This is an entry for a module of the type password. It is used when a user attempts to change the password. In this case, the module `pam_pwcheck.so` is used to check if a new password is secure enough.

The **nullok** argument allows users to change an empty password, otherwise empty passwords are treated as locked accounts.

- **password required pam\_unix2.so nullok use\_first\_pass use\_authtok**

The `pam_unix2.so` module is also necessary when changing a password. It encrypts (or hashes, to be more exact) the new password, and writes it to the authentication database.

**nullok** has the same significance as described above for `pam_pwcheck.so`. With the argument **use\_first\_pass**, `pam_unix2` uses the password from a previous module, for instance `pam_pwcheck.so`, and aborts with an error if no authentication token from a previous module is available. The argument **use\_authtok** is used to force this module to set the new password to the one provided by the previously stacked password module.

- **session required pam\_unix2.so**

Here the session component of the `pam_unix2.so` module is used. Without arguments this module has no effect, with the argument **trace** it uses the syslog daemon to log the user's login.

- **session required pam\_limits.so**

The `pam_limits.so` sets resource limits for the users that can be configured in the file `/etc/security/limits.conf`.

- **session required pam\_mail.so**

This module displays a message if any new mail is in the user's mail box. It also sets an environment variable pointing to the user's mail directory.

## ***Secure Password Guidelines***

Even the best security setup for a system can be defeated if users choose easy to guess passwords. With today's computing power, a simple password can be cracked within minutes.

These attacks are also called *dictionary attacks*, as the password cracking program just tries one word after another from a dictionary file, including some common variations of these words.

Therefore, a password should never be a word which could be found in a dictionary. A good, secure password should always contain numbers and uppercase characters.

To check whether user passwords fulfill this requirement, you can enable a special PAM module to test a password first before a user can set it. The PAM module is called `pam_pwcheck.so` and uses the `cracklib` library to test the security of passwords.

By default, this PAM module is enabled on SLES 10.

If a user enters a password that is not secure enough, the following message is displayed:

**Bad password: too simple**

and the user is prompted to enter a different one.

There are also dedicated password check programs available such as **John the Ripper** (<http://www.openwall.com/john/>).

## ***PAM Documentation Resources***

The following PAM documentation is available in the directory `/usr/share/doc/packages/pam/`:

- **READMEs.** In the top level of this directory, there are some general README files. The subdirectory `modules/` holds README files about the available PAM modules.
- **The Linux-PAM System Administrators' Guide.** This document includes everything that a system administrator should know about PAM.

The document discusses a range of topics, from the syntax of configuration files to the security aspects of PAM. The document is available as a PDF file, in HTML format, and as plain text.

- **The Linux-PAM Module Writers' Manual.** This document summarizes the topic from the developer's point of view, with information about how to write standard-compliant PAM modules. It is available as a PDF file, in HTML format, and as plain text.
- **The Linux-PAM Application Developers' Guide.** This document includes everything needed by an application developer who wants to use the PAM libraries. It is available as a PDF file, in HTML format, and as plain text.

There are also manual pages for some PAM modules, such as **`man pam_unix2`**.



## **Exercise 6-1      Configure PAM Authentication**

In this exercise, you practice configuring PAM authentication. Create the file `/etc/nologin` to prevent all normal users (such as `geeko`) from logging in. Try to login as `geeko` on a text console. As root, modify `/etc/pam.d/login` by putting a `#`-sign in front of the line with `pam_nologin.so`. Again try to login as `geeko` on a text console. Undo your changes in `/etc/pam.d/login` and remove the file `/etc/nologin`.

### **Detailed Steps to Complete this Exercise:**

1. From the graphical desktop, switch to virtual console 3 (**Ctrl+Alt+F3**); then log in as **root** (password **novell**).
2. Create the file `/etc/nologin` by entering  
**echo No login possible > /etc/nologin**.
3. Switch to virtual console 4 (**Alt+F4**).
4. Attempt to log in as **geeko** (password **novell**).  
“No login possible” plus a “Login incorrect” message are displayed, indicating that you cannot log in to the system.
5. Switch back to virtual console 3 (**Alt+F3**).
6. View the last lines of the file `/var/log/messages` by entering the following:  
**tail /var/log/messages**  
Look for the FAILED LOGIN message for `geeko` that indicates the failed login attempt.
7. Edit the file `/etc/pam.d/login`:
  - a. Enter **vi /etc/pam.d/login**.
  - b. Switch to the text insert mode by pressing **Insert**.

- c. Add a # sign to the beginning of the following line, as in the following:

**#auth required pam\_nologin.so**

This PAM module is required to be successful during system authentication. It checks to see if the file `/etc/nologin` exists, and if it does, this PAM module does not allow regular users to log in by returning a failed status.

Now that this line is commented out, PAM will not check for the file. This means that all users can log in, even if the file exists.

- d. Return to the command mode by pressing **Esc**; save the file by entering **:w**.
- 8. Test the modified PAM configuration file:
    - a. Switch to virtual console 4 (**Alt+F4**).
    - b. Attempt to log in as **geeko** (password **novell**).

You are able to log in because PAM no longer checks for the file `/etc/nologin`.
    - c. Logout as **geeko** by entering **exit**.
  - 9. Edit the file `/etc/pam.d/login` to uncomment the `pam_nologin.so` line:
    - a. Switch to virtual console 3 (**Alt+F3**).
    - b. Uncomment the `pam_nologin.so` line (by removing the # sign you entered before) so it looks like the following:

**auth required pam\_nologin.so**
    - c. Return to the command mode by pressing **Esc**; save the file and exit vi by entering **:wq**.
  - 10. On virtual console 4, try logging in as **geeko**.

Again you will receive a “Login incorrect” message.
  - 11. As **root** on virtual console 3 delete the file `/etc/nologin` by entering  
**rm /etc/nologin**.

**12.** On virtual console 4, try again to log in as **geeko**.

Because the file `/etc/nologin` does not exist, login for normal users is enabled again.

**13.** Log out as **geeko** by entering **exit**.

**14.** Switch to virtual console 3 and log out as root by entering **exit**.

**15.** Return to the desktop by pressing **Alt+F7**.

***(End of Exercise)***

## Objective 2    **Configure Security Settings**

YaST provides a Local Security module that lets you configure the following local security settings for your SUSE Linux Enterprise Server:

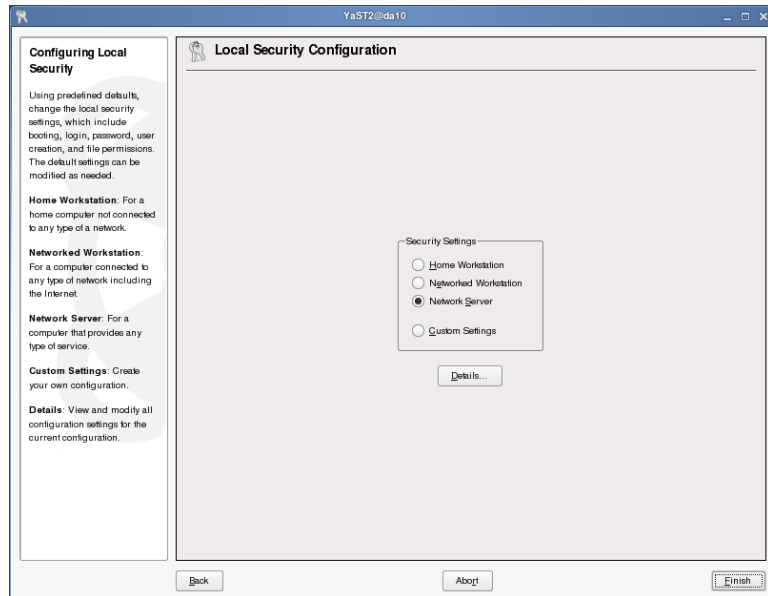
- Password settings
- Boot configuration
- Login settings
- User creation settings
- File permissions

You can select from (or modify) three preset levels of security, or create your own customized security settings to meet the requirements of your enterprise security policies and procedures.

You can access the Security Settings module from the YaST Control Center by selecting **Security and Users > Local Security**, or by entering as root **yast2 security** in a terminal window.

The following appears:

**Figure 6-3**



From this dialog, you can select one of the following preset configurations:

- **Home Workstation.** Select for a home computer not connected to any type of a network. This option represents the lowest level of local security.
- **Networked Workstation.** Select for a computer connected to any type of a network or the Internet. This option provides an intermediate level of local security.
- **Network Server.** Select for a computer that provides any type of service (network or otherwise). This option enables a high level of local security.
- You can also select **Details** or **Custom Settings** to modify an existing security level or create your own configuration.

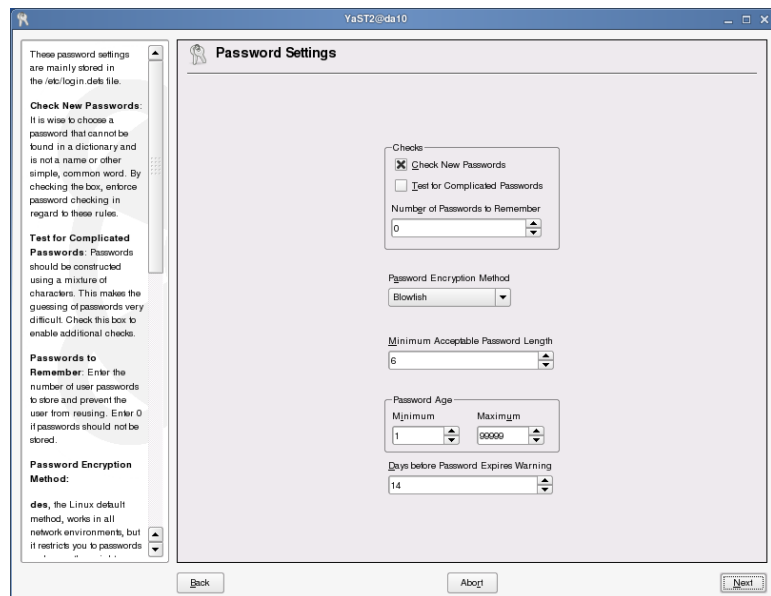
By selecting one of the three predefined security levels and selecting **Next**, the chosen security level is applied. By selecting **Details**, you can change the settings for the security level you have selected.

If you choose the **Customs Settings** and then select **Next**, you can directly change the details of the security configuration.

The dialogs for the detail settings look the same for every security level, but the preselected options are different. In the following dialogs, you see the settings for Level 3 (Network Server).

In the first dialog you can change the default password requirements that are accepted by the systems:

**Figure 6-4**



From this dialog, you can select or enter the following password settings (mainly stored in `/etc/login.defs`, some values also in `/etc/default/passwd` and `/etc/security/pam_pwcheck.conf`):

- **Check New Passwords.** It is important to choose a password that cannot be found in a dictionary and is not a name or other simple, common word. By selecting this option, you enforce password checking in regard to these rules.
- **Test for Complicated Passwords.** Passwords should be constructed using a mixture of uppercase and lower case characters as well as numbers. Special characters like ;(= etc. may be used too, but could be hard to enter on a different keyboard layout. This makes it very difficult to guess the password. Select this option to enable additional checks.
- **Password Encryption Method.** From the drop-down list, select one of the following encryption methods:
  - **DES.** This is the lowest common denominator. It works in all network environments, but it restricts you to passwords no longer than eight characters. If you need compatibility with other systems, select this method.
  - **MD5.** This encryption method allows longer passwords and is supported by all current Linux distributions, but not by other systems or older software.
  - **Blowfish.** This encryption method uses the blowfish algorithm to encrypt passwords. It is not yet supported by many systems. A lot of CPU power is needed to calculate the hash, which makes it difficult to crack passwords with the help of a dictionary. It is used as default encryption method on SLES 10
- **Minimum Acceptable Password Length.** Enter the minimum number of characters for an acceptable password. If a user enters fewer characters, the password is rejected.

Entering **0** disables this check.
- **Password Age.** Minimum refers to the number of days that have to elapse before a password can be changed again. Maximum is the number of days after which a password expires and must be changed.

- **Days Before Password Expires Warning.** A warning is issued to the user this number of days before password expiration.

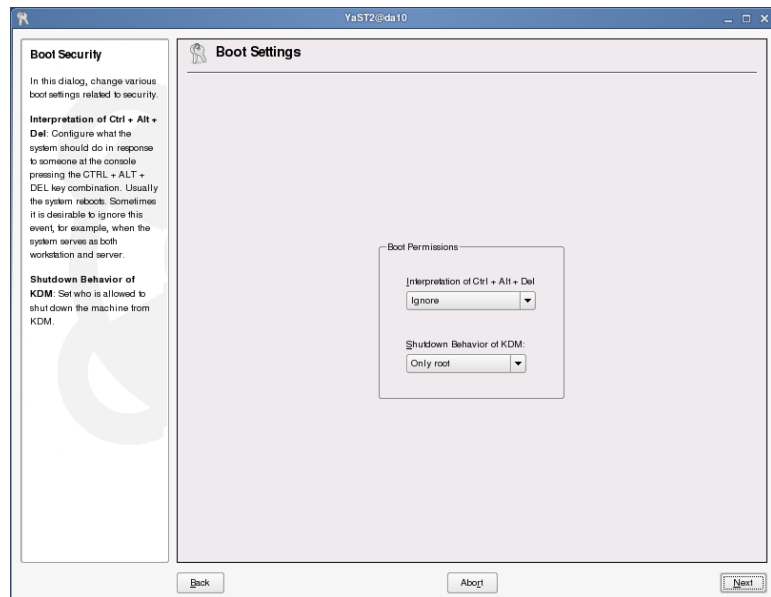


Although root receives a warning when setting a password, she can still enter a bad password despite the above settings.

---

When you finish configuring password settings, continue by selecting **Next**. The following appears:

**Figure 6-5**



From this dialog, you can select the following boot settings (which update the file `/etc/inittab`):

- **Interpretation of Ctrl + Alt + Del.** When someone at the console presses the **Ctrl+Alt+Del** keystroke combination, the system usually reboots.

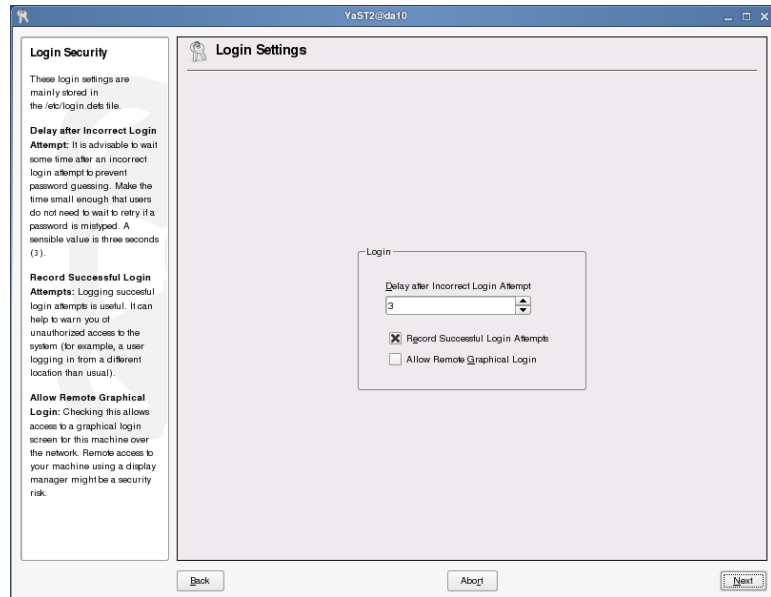


- ❑ **Ignore.** Sometimes you want to have the system ignore this keystroke combination, especially when the system serves as both workstation and server. Nothing happens when the **Ctrl+Alt+Del** keystroke combination is pressed.
- ❑ **Reboot.** The system reboots when the **Ctrl+Alt+Del** keystroke combination is pressed.
- ❑ **Halt.** The system is shut down when the **Ctrl+Alt+Del** keystroke combination is pressed.
- **Shutdown Behavior of KDM.** You use this option to set who is allowed to shut down the computer from KDM.
  - ❑ **Only Root.** To halt the system, the root password has to be entered.
  - ❑ **All Users.** Everyone, even remotely connected users, can halt the system using KDM.
  - ❑ **Nobody.** Nobody can halt the system with KDM.
  - ❑ **Local Users.** Only locally connected users can halt the system with KDM.
  - ❑ **Automatic.** The system is halted automatically after log out.

For a server system you should use **Only Root** or **Nobody** to prevent normal or even remote users from halting the system

When you finish configuring boot settings, continue by selecting **Next**. The following appears:

**Figure 6-6**



From this dialog, you can enter and select the following login settings (mainly stored in `/etc/login.defs`):

- **Delay After Incorrect Login Attempt.** Following a failed login attempt, there is typically a waiting period of a few seconds before another login is possible. This makes it more difficult for password crackers to log in.

This option lets you adjust the time delay before another login attempt. Default is 3 seconds, which is a reasonable value.

- **Record Successful Login Attempts.** Recording successful login attempts can be useful, especially in warning you of unauthorized access to the system (such as a user logging in from a different location than normal).

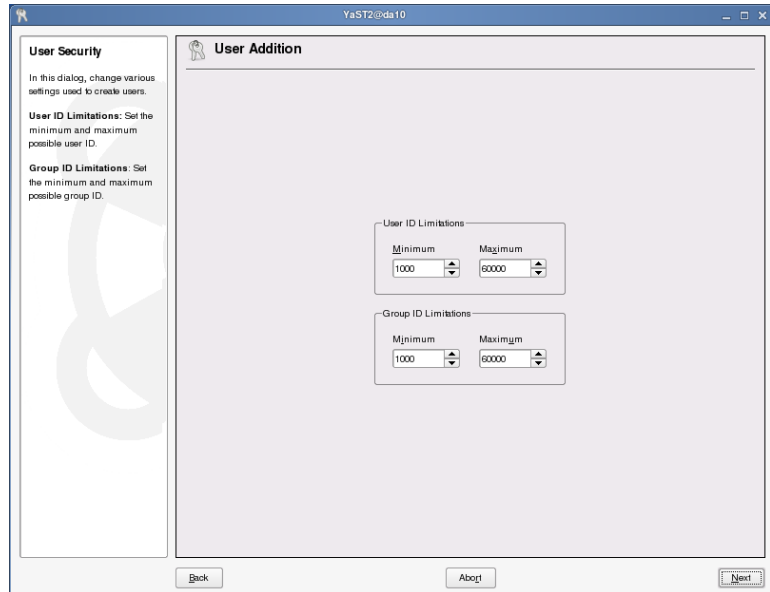
Select this option to record successful login attempts in the file **/var/log/wtmp**. You can use the command **last** to view who logged in at what time.

- **Allow Remote Graphical Login.** You can select this option to allow other users access to your graphical login screen via the network.

Because this type of access represents a potential security risk, it is inactive by default.

When you finish configuring login settings, continue by selecting **Next**. The following appears:

**Figure 6-7**

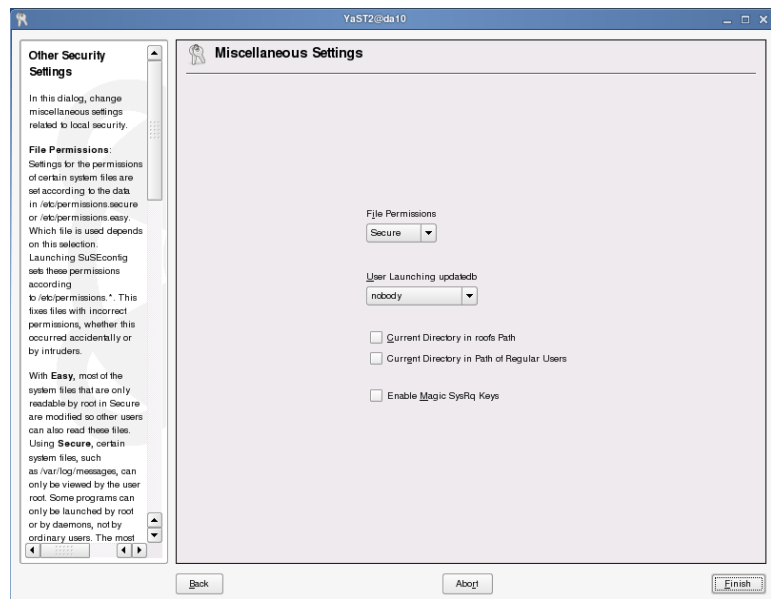


From this dialog, you can enter the following ID settings (stored in **/etc/login.defs**):

- **User ID Limitations.** Enter a minimum and maximum value to configure a range of possible user ID numbers. New users get a UID from within this range.
- **Group ID Limitations.** Enter a minimum and maximum value to configure a range of possible group ID numbers.

When you finish configuring user and group ID limitations, continue by selecting **Next**. The last page of the security configuration appears:

**Figure 6-8**



From this dialog, you can select the following miscellaneous global settings:

- **File Permissions.** Settings for the permissions of certain system files are configured in **/etc/permissions.easy**, **/etc/permissions.secure**, or **/etc/permissions.paranoid**. You can also add your own rules to the file **/etc/permissions.local**. Each file contains a description of the file syntax and purpose of the preset.

Settings in files in the directory **/etc/permissions.d/** are included as well. This directory is used by packages that bring their own permissions files.

From the drop-down list, select one of the following:

- **Easy.** Select this option to allow read access to most of the system files by users other than root.
- **Secure.** Select this option to make sure that certain configuration files (such as **/etc/ssh/sshd\_config**) can only be viewed by the user root. Some programs can only be launched by root or by daemons, not by an ordinary user.
- **Paranoid.** Select this option for an extremely secure system. All SUID/SGID-Bits on programs have been cleared. Remember that some programs might not work correctly, because users no longer have the permissions to access certain files.

Running SuSEconfig sets these permissions according to the settings in the respective **/etc/permissions\*** files. This fixes files with incorrect permissions, whether this occurred accidentally or by intruders.

- **User Launching updatedb.** If the program **updatedb** is installed, it automatically runs on a daily basis or after booting. It generates a database (**locatedb**) in which the location of each file on your computer is stored.

You can search this database with the utility **locate** (enter **man locate** for details).

From the drop-down list, select one of the following:

- ❑ **nobody.** Any user can find only the paths in the database that can be seen by any other (unprivileged) user.
- ❑ **root.** All files in the system are added into the database.
- **Current Directory in root's Path and Current Directory in the Path of Regular Users.**

If you deselect these options (the default), users must always launch programs in the current directory by adding “./” (such as **./configure**).

If you select these options, the dot (“.”) is appended to the end of the search path for root and users, allowing them to enter a command in the current directory without appending “./”.

Selecting these options can be very dangerous because users can accidentally launch unknown programs in the current directory instead of the usual system-wide files.

This configuration is written to **/etc/sysconfig/suseconfig**.

- **Enable Magic SysRq Keys.** Selecting this option gives you some control over the system even if it crashes (such as during kernel debugging). For details, see [/usr/src/linux/Documentation/sysrq.txt](#).

This configuration is written to **/etc/sysconfig/sysctl**.

When you finish configuring the miscellaneous settings, save the settings and run SuSEconfig by selecting **Finish**.

## **Exercise 6-2      Configure the Password Security Settings**

In this exercise, you practice changing different security settings. Change the default behavior when the keys Ctrl+Alt+Del are pressed to halting the machine. Also change the encryption from blowfish to MD5. Use the YaST Local Security module to do the above.

### **Detailed Steps to Complete this Exercise:**

1. Open a terminal window.
2. Check the setting for the Ctrl+Alt+Del keystroke in the file `/etc/inittab` by entering **grep ctrlaltdel /etc/inittab**.  
Note the current setting:
3. Start **YaST** and select **Security and Users > Local Security**.  
The Local Security Configuration dialog appears.
4. Make sure **Custom Settings** is selected; then select **Next**.  
The Password Settings dialog appears.
5. From the Password Encryption Method drop-down list, select **MD5**.
6. Continue by selecting **Next**.  
The Boot Settings dialog appears.
7. From the Interpretation of Ctrl + Alt + Del drop-down list, select **Halt**.
8. Continue by selecting **Next**.  
The Login Settings dialog appears.

9. Accept the default settings by selecting **Next**.

The Adding User dialog appears.

10. Accept the default settings by selecting **Next**.

The Miscellaneous Settings dialog appears.

11. Accept the default settings and configure the system for the new settings by selecting **Finish**.

12. To test the change, you must first activate the new configuration by rebooting the system or by entering (as root) **init q** (reload the `/etc/inittab` file) in a terminal window.

- a. From the terminal window, **su -** to root (password **novell**).

- b. Reload the file `/etc/inittab` by entering **init q**.

13. Verify that the **Ctrl+Alt+Del** setting has changed by entering **grep ctrlaltdel /etc/inittab**.

Notice that the setting is now “shutdown -h” instead of what you noted in Step 2.

14. Test this setting by pressing **Ctrl+Alt+F2**; then press **Ctrl+Alt+Del**.

The system shuts down instead of restarting.

15. Turn on your computer and log in to the Gnome desktop as **geeko**.

16. (Optional) Use the YaST Security settings module to change the default for **Ctrl+Alt+Del** back to restart.

**(End of Exercise)**



# Summary

Objective	Summary
1. Configure User Authentication with PAM	<p>Linux uses PAM (Pluggable Authentication Modules) in the authentication process as a layer that communicates between users and applications.</p> <p>Within the PAM framework there are four different module types: auth, account, session, and password. Control flags—required, requisite, sufficient, optional—govern what happens on success or failure of a module.</p> <p>Files in /etc/pam.d/ are used to configure PAM, with additional configuration options in files in /etc/security/ for certain modules.</p>
2. Configure Security Settings	<p>Defaults for user accounts and other security relevant settings can be configured using the YaST Local Security module.</p> <p>The configuration settings are written to various files, the most pertinent being files in /etc/default/, and /etc/login.defs.</p>



## SECTION 7 Use Syslog Daemon syslog-ng

Up to SUSE Linux Enterprise Server 9, syslogd was used to log system events. With SUSE Linux Enterprise Server 10 these events are logged by syslog-ng, the new generation syslogd.

The main advantage of syslog-ng over syslogd is its capability to filter messages not only based on facilities and priorities, but also based on the content of each message.

### Objectives

1. Use Syslog Daemon syslog-ng

## Objective 1      Use Syslog Daemon **syslog-ng**

The syslog daemon **syslog-ng** is used by many services to log system events. The advantage in using a single service for logging is that all logging can be managed from one configuration file.

The syslog daemon accepts messages from system services, and, depending on its configuration, other hosts, and logs them based on settings in the configuration files **/etc/sysconfig/syslog** and **/etc/syslog-ng/syslog-ng.conf**. The file **/etc/syslog-ng/syslog-ng.conf** is generated by SuSEconfig from **/etc/syslog-ng/syslog-ng.conf.in**. Both files share the same syntax.

The configuration of syslog-ng is distributed across three files:

- **/etc/sysconfig/syslog**
- **/etc/syslog-ng/syslog-ng.conf.in**
- **/etc/syslog-ng/syslog-ng.conf**

## ***/etc/sysconfig/syslog***

The file **/etc/sysconfig/syslog** contains general parameters applicable to syslog-ng as well as syslogd.

Parameters set in this file include switches passed to syslogd or syslog-ng, kernel log level, parameters for klogd, and which syslog daemon is to be used.

```
...
## Type:                string
## Default:             " "
## Config:              " "
## ServiceRestart:     syslog
#
# if not empty: parameters for syslogd
# for example SYSLOGD_PARAMS="-r -s my.dom.ain"
#
SYSLOGD_PARAMS=" "

## Type:                string
## Default:             -x
## Config:              " "
## ServiceRestart:     syslog
#
# if not empty: parameters for klogd
# for example KLOGD_PARAMS="-x" to avoid (duplicate) symbol
resolution
#
KLOGD_PARAMS="-x"

## Type:                list(syslogd,syslog-ng)
## Default:             syslogd
## Config:              syslog-ng
## Command:             /sbin/rcsyslog restart
## PreSaveCommand:     /sbin/rcsyslog status &&
                        /sbin/rcsyslog stop
#
# The name of the syslog daemon used as
# syslog service: "syslogd", "syslog-ng"
#
SYSLOG_DAEMON="syslog-ng"
...
```

Parameters set in `/etc/sysconfig/syslog` are evaluated by the start script `/etc/init.d/syslog`. Furthermore, `SuSEconfig` uses `/etc/sysconfig/syslog` to add log sockets to the file `/etc/syslog-ng/syslog-ng.conf` when generating this file from `/etc/syslog-ng/syslog-ng.conf.in`.

### ***/etc/syslog-ng/syslog-ng.conf.in***

**`/etc/syslog-ng/syslog-ng.conf.in`** is the template used to create the configuration file **`/etc/syslog-ng/syslog-ng.conf`**, which is the configuration file actually used by `syslog-ng`. Both files have the same syntax.

However, unless you turn off generation of `/etc/syslog-ng/syslog-ng.conf` in `/etc/sysconfig/syslog`, any manual changes to this file will be overwritten when `SuSEconfig` is executed.

Therefore, changes to the configuration of `syslog-ng` should be made in this file.

### ***/etc/syslog-ng/syslog-ng.conf***

`syslogd` and `syslog-ng` share two concepts that you have to understand to be able to configure either one:

- Facilities
- Priorities

The configuration of `syslog-ng` consists of several parts which are then combined to configure which information is logged where.

These are:

- Sources
- Filters

- Destinations
- Log Paths

## Facilities

The facility refers to the subsystem that provides the corresponding message. Each program that uses syslog for logging is assigned such a facility, usually by its developer.

The following describes these facilities:

**Table 7-1**

Facility	Description
authpriv	Used by all services that have anything to do with system security or authorization. All PAM messages use this facility.  The ssh daemon uses the auth facility.
cron	Accepts messages from the cron and at daemons.
daemon	Used by various daemons that do not have their own facility, such as the ppp daemon.
kern	All kernel messages.
lpr	Messages from the printer system.
mail	Messages from the mail system. This is important because many messages can arrive very quickly.
news	Messages from the news system. As with the mail system, many messages might need to be logged in a short time.
syslog	Internal messages of the syslog daemon.

**Table 7-1** *(continued)*

Facility	Description
user	A general facility for messages on a user level. For example, It is used by login to log failed login attempts.
uucp	Messages from the uucp system.
local0 – local7	<p>These 8 facilities are available for your own configuration. All of the local categories can be used in your own programs.</p> <p>By configuring one of these facilities, messages from your own programs can be administered individually through entries in the file <code>/etc/syslog-ng/syslog-ng.conf</code>.</p>

## Priorities

The priority gives details about the urgency of the message. The following priorities are available (listed in increasing degree of urgency):

**Table 7-2**

Priority	Description
debug	Should only be used for debugging purposes, since all messages of this category and higher are logged.
info	Used for messages that are purely informative.
notice	Used for messages that describe normal system states that should be noted.
warning	Used for messages displaying deviations from the normal state.
err	Used for messages displaying errors.
crit	Used for messages on critical conditions for the specified program.



**Table 7-2** *(continued)*

Priority	Description
alert	Used for messages that inform the system administrator that immediate action is required to keep the system functioning.
emerg	Used for messages that warn you that the system is no longer usable.

### Sources

A source is a collection of source drivers, which collect messages using a given method. These sources are used to gather log messages. The general syntax is as follows:

```
source <identifier> { source-driver(params); source-driver(params); ... };
```

The respective section in `/etc/syslog-ng/syslog-ng.conf` looks like this:

```
source src {
    # include internal syslog-ng messages
    # note: the internal() source is required!
    internal();

    # the following line will be replaced by the
    # socket list generated by SuSEconfig using
    # variables from /etc/sysconfig/syslog:
    unix-dgram("/dev/log");

    # uncomment to process log messages from network:
    #udp(ip("0.0.0.0") port(514));
};
```

In this example, one source for internal messages of syslog-ng and the socket `/dev/log` are defined.

## Filters

Filters are boolean expressions that are applied to messages and are evaluated as either true or false. The general syntax is as follows:

```
filter <identifier> { expression; };
```

The identifier has to be unique within the configuration and is used later to configure the actual logging.

The following excerpt of `/etc/syslog-ng/syslog-ng.conf` shows some filters used in SUSE Linux Enterprise Server 10:

```
#
# Filter definitions
#
filter f_iptables    { facility(kern) and match("IN=") and match("OUT=");
};

filter f_console     { level(warn) and facility(kern) and not
                      filter(f_iptables) or level(err) and not facility(authpriv); };

filter f_newsnotice  { level(notice) and facility(news); };
filter f_newscrit    { level(crit)   and facility(news); };
filter f_newserr     { level(err)    and facility(news); };
filter f_news        { facility(news); };
...
filter f_messages    { not facility(news, mail)
                      and not filter(f_iptables); };
...
```

As you can see, facility and priority (level) can be used within filters. However, it is also possible to filter according to the content of a line being logged, as in the `f_iptables` filter above.

Combining the expressions with “and”, “or”, or “and not” allows you to create very specific filters.

## Destinations

Destinations defines where messages can be logged. The general syntax is as follows:

```
destination <identifier> {  
    destination-driver(params);  
    destination-driver(params); ... };
```

Possible destinations are files, fifos, sockets, ttys of certain users, programs, or other hosts.

A sample from `/etc/syslog-ng/syslog-ng.conf` looks like this:

```
destination console { file("/dev/tty10"    group(tty) perm(0620)); };  
destination messages { file("/var/log/messages"); };
```

## Log Paths

Log paths are the point where it all comes together. They define which messages are logged where, depending on source, filter, and destination. The general syntax is as follows:

```
log { source(s1); source(s2); ...  
      filter(f1); filter(f2); ...  
      destination(d1); destination(d2); ...  
      flags(flag1[, flag2...]); };
```

The following entries in `/etc/syslog-ng/syslog-ng.conf` for instance are responsible for logging to `/dev/tty10` and `/var/log/messages`:

```
log { source(src); filter(f_console); destination(console); };  
log { source(src); filter(f_messages); destination(messages); };
```

In the first line, log messages that come in through sources defined in source `src` are logged to `tty10` if they match the filter `f_console`. In line two, messages that come in through sources defined in source `src` are logged to `/var/log/messages` if they match the filter `f_messages`.



---

For further details on the `syslog-ng.conf` file, enter **`man 5 syslog-ng.conf`**. The documentation in `/usr/share/doc/packages/syslog-ng/html/book1.html` gives a general overview of `syslog-ng` as well as details on the configuration.

---

## Summary

Objective	Summary
1. Use Syslog Daemon syslog-ng	<p>In a Linux system, there are many logs that track various aspects of system operation. Many services log their activities to their own log files, and the level of detail can be set on a per-service basis. In addition, system logs in /var/log/ track system-level events.</p> <p>logrotate is the utility to archive log files.</p>

---



## SECTION 8    Manage Virtualization with Xen

Virtualization is one of the hottest topics in the industrie at the moment. However, the idea of virtualization is not new at all. Hardware platforms like IBMs pSeries or zSeries support virtualization since a long time and software like VMware Workstation for x86 based systems has been available for many years.

Now virtualization moves to mainstream, because affordable Intel or AMD based x86 systems, provide enough resources to run more than one virtual machine at the same time.

SUSE Linux Enterprise Server 10 comes with build-in virtualization support through the Xen virtual machine monitor. In the following section you'll learn how to use this powerful feature.

In this section you learn about the Xen virtualization technology in SUSE Linux Enterprise Server 10.

### Objectives

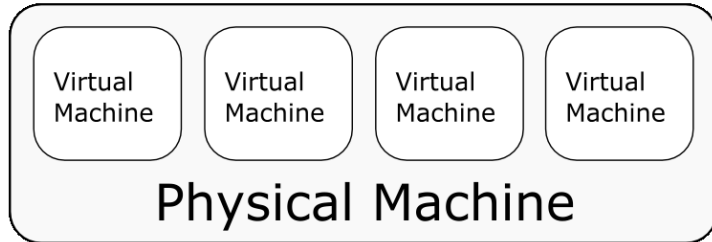
1. Understand the Concept of Virtualization
2. Understand How Xen Works
3. Install Xen
4. Manage Xen Domains with YaST
5. Manage Xen Domains at the Command Line
6. Understand Xen Networking
7. Migrate a Guest Domain

## Objective 1 Understand the Concept of Virtualization

Virtualization technology separates a running instance of an operating system from the physical hardware. Instead of a physical machine, the operating system runs in a so-called virtual machine. Multiple virtual machines share the resources of the underlying hardware.

Virtualization allows you to run multiple virtual systems on one single physical machine.

**Figure 8-1**



The following are the main advantages of virtualization, in comparison with non-virtualized physical hardware:

1. **Efficient Hardware Utilization.** Often systems are not using the full potential of their hardware. By running multiple virtual machines on the same hardware, the resources are used more efficiently.
2. **Reduced Downtime.** Virtual machines can be easily migrated to to a new physical host system. This reduces the downtime in case of a hardware failure.
3. **Flexible Resource Allocation.** Hardware resources can be allocated on demand. When the resource requirements of a virtual machine change, resource allocation can be adjusted or the machine can be migrated to a different physical host.



## Objective 2      Understand How Xen Works

The idea of virtualization is not new. Platforms like IBM zSeries or pSeries offer built-in virtualization and Intel x86 based systems can be virtualized using third-party software like VMware.

SUSE Linux Enterprise Server 10 comes with a virtualization technology called Xen, which allows you to run multiple virtual machines on a single piece of Intel x86 based hardware.

At the moment, the operating systems that run in a Xen virtual machine need to be modified. Therefore only open source operating systems like Linux or BSD can be installed. One exception is Netware, which has been adjusted by Novell to run in a Xen virtual machine.

Intel and AMD are developing extensions (Intel Vanderpool and AMD Pacifica) to the x86 Standard to support virtualization. Once these extensions are available, Xen will be able to run unmodified operating systems including Microsoft Windows.



---

You can find updated information about Xen, including an instruction how to run unmodified operating systems on the OpenSUSE Xen page at: **<http://en.opensuse.org/Xen>**

---

To understand how Xen works, you need to do the following:

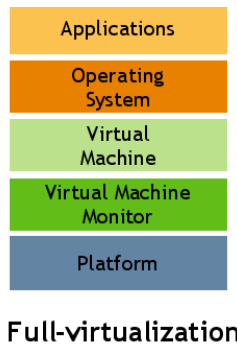
- Understand Virtualization Methods
- Understand the Xen Architecture

## ***Understand Virtualization Methods***

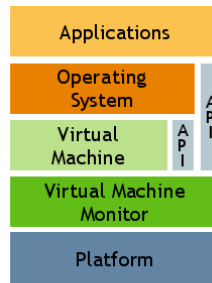
Before we talk in detail about the Xen technology, you should understand the following two different virtualization methods.

- **Full Virtualization.** In this case the virtualization software emulates a full virtual machine including all hardware resources. The operating system running in the virtual machine (guest OS) communicates with these resources as if they were physical hardware. VMware Workstation is a popular full virtualization software.

**Figure 8-2**



- **Para Virtualization.** Instead of emulating a full virtual machine, para-virtualization software provides an Application Programming Interface (API) which is used by the guest OS to access hardware resources. This requires that the guest OS is aware that it runs in a virtual machine and needs to know how to access the API. Xen is a para-virtualization software.

**Figure 8-3**

### Para-virtualization

Para virtualization provides better performance because it does not emulate all hardware details. The drawback is, that the guest OS needs to be modified to run with para-virtualization.

Full-virtualization works with an unmodified guest OS but generates more overhead resulting in a weaker performance.

Another advantage of para-virtualization is the flexible resource allocation. As the guest OS is aware of the virtual environment, Xen can, for example, change the memory allocation of a virtual machine on the fly without any reboot.

## ***Understand the Xen Architecture***

Xen consists of the following two major components:

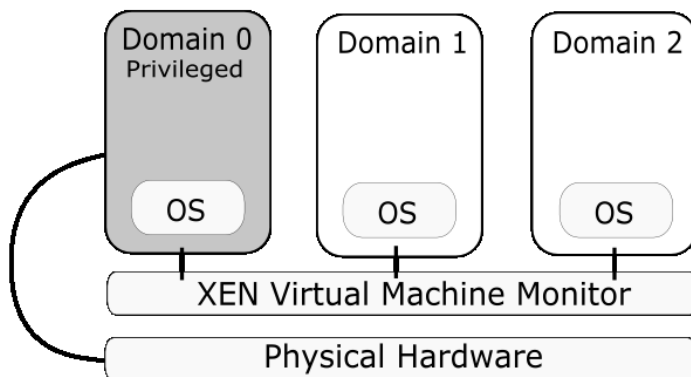
- **Virtual Machine Monitor.** The virtual machine monitor forms a layer between physical hardware and virtual machines. In general this kind of software is called a **Hypervisor**.
- **Xen tools.** The Xen tools are a set of command line applications that are used to administer virtual machines.

The virtual machine monitor must be loaded before any of the virtual machines are started. When working with Xen, virtual machines are called **domains**.

The Xen virtual machine monitor neither includes any drivers to access the physical hardware of the host machine nor an interface to communicate directly with an administrator. These tasks are performed by an operating system running in the privileged **domain0**.

The following is an overview of a Xen system with three domains.

**Figure 8-4**



A process called **xend** runs in the domain0 Linux installation. This process is used to manage all Xen domains running on a system and to provide access to their consoles.

A unprivileged domain is also called domainU in the Xen terminology.

SUSE Linux Enterprise Server 10 can be used for privileged (domain0) and unprivileged (domainU) Xen domains.

## Objective 3    Install Xen

To setup a Xen system, you start from a normal SUSE Linux Enterprise 10 installation, which is going to run in domain0.

The other Xen domains can later be installed in physical partitions or file system images. When you plan to use physical partitions, you have to make sure that the initial SUSE Linux Enterprise Server 10 installation is not using all of the available disc space.

For maximum flexibility it makes sense to use the logical volume manager LVM or EVMS for a Xen system.

The following packages have to be installed in the initial SUSE Linux Enterprise Server 10 installation:

- **xen.** This package contains the Xen virtual machine monitor (Hypervisor).
- **xen-tools.** Contains xend and a collection of command line tools to administer a Xen system.
- **kernel-xen.** This package contains a modified Linux kernel that runs in a Xen domain.
- **xen-doc-\*** (optional). Xen documentation in various formats.

The installation of the Xen package automatically adds an entry like the following into the bootloader configuration file **/boot/grub/menu.lst**.

```
title Xen
root (hd0,3)
kernel /boot/xen.gz
module /boot/vmlinuz-Xen root=/dev/hda3 selinux=0
module /boot/initrd-Xen
```



On some Xen systems you might see the parameter **dom0\_mem** in the kernel module line. This parameter assigns a certain amount of initial main memory to domain0 at boot time. However in Xen version 3, this parameter is not required anymore.

Initially all available memory is used by domain0. When you start additional domainUs, the required amount of memory is reduced in domain0 and used for the new domainU.

---

The entry in menu.lst adds a new option to the boot menu of your system. When selecting this entry, the Xen virtual machine monitor is loaded (**kernel /boot/xen.gz**) which starts SUSE Linux Enterprise Server 10 in domain0 (see the lines starting with **module**).

Before rebooting your system with the Xen option, you should check if the automatically generated entry is correct. Make sure that ...

- ... the line **root (hd0,3)** points to the filesystem which contains the Xen Virtual Machine Monitor and the Kernel of the Linux installation for domain0. In our example **hd0,3** means the fourth partition on the first hard drive in the system. Also check if the parameter **root** in the first module line points to the root partition of the domain0 installation.
- ... the Xen version of the Linux kernel and the initrd are loaded in the module line. The names of the image files should end in **-xen**.

After checking the bootloader configuration file, you can reboot your system and select the Xen option at the bootloader menu. In the early stages of the boot process, you will see some messages of the Xen virtual machine monitor on the screen. Then the domain0 Linux installation is started.

In case the system is not booting properly, you can switch back to a non-virtualized system by selecting the regular SUSE Linux Enterprise Server 10 boot option.



---

When running Xen, the network setup is done by the xend management process. This can interfere with the native network configuration scripts of the domains. Especially SuSEfirewall2 is known to cause problems. It's therefore recommended to stop SuSEfirewall2 with **rcSuSEfirewall2** and to remove the firewall scripts from the init process:

```
insserv -r SuSEfirewall2_setup  
insserv -r SuSEfirewall2_init  
insserv -r SuSEfirewall2_final (conditional)
```

---



## **Exercise 8-1     *Install Xen***

In this exercise, you learn how to install Xen and configure domain0.

Do the following:

- Part I: Install XenPackages.
- Part II: Prepare for Reboot
- Part III: Reboot and Test Xen.

### **Part I: Install XenPackages.**

Do the following:

1. Start the **YaST Controll Center**.
2. Select **Software > Software Management**.
3. From the Filter menu, select **Search**.
4. Enter **xen** in the search field and select **search**.
5. On the right side, select the packages **xen**, **kernel-xen** and **xen-tools**.
6. Select **Accept** and let YaST install all required software packages.
7. **Close** the YaST Control Center.

### **Part II: Prepare for Reboot**

Do the following:

1. Open a terminal window and **su -** to the root user.
2. Open the file **/boot/grub/menu.lst** with a text editor (eg. vi).
3. Make sure, that there is a section with the title **Xen** in the file.

4. In this section, make sure that the parameter **root=** points to the root partition of your installation.
5. Close the file.
6. Enter the command:  
**insserv -r SuSEfirewall2\_setup**  
and  
**insserv -r SuSEfirewall2\_init**
7. Close the terminal window.

### **Part III: Reboot and Test Xen.**

1. Reboot your system.
2. At the boot menu, select the **Xen entry** and hit **Return**.
3. When the system has been booted, log in as user **geeko** with the password **novell**.
4. Open a terminal window and **su -** to the **root** user.
5. Enter the command **xm list**.
6. In the output you should see one domain (Domain-0) with the status running.

***(End of Exercise)***

## Objective 4    Manage Xen Domains with YaST

After you have installed Xen and the Xen tools, you can start to create more Xen domains. Before we go into the details of the domain configuration, we will introduce the YaST module **Virtual Machine Management (Xen)**.

This module provides a convenient way to create and control the Xen domains on your system. The module can be started from the **System** section in the YaST Control Center, and has to run on the Linux system running in domain0.

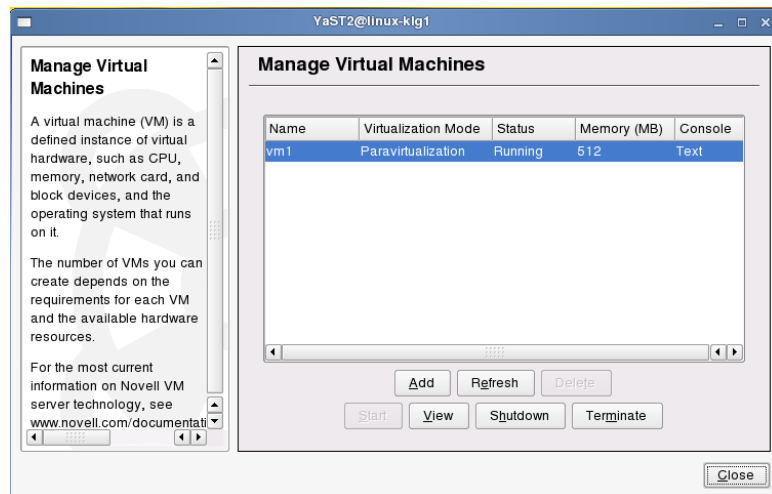


Not every detail of the Xen domain configuration is described in the following. More in-depth information follow in the next objective.

The following is a step by step description of how to create and boot a new Xen domain with this YaST module.

After you have started the module, the following dialog appears on the screen:

**Figure 8-5**



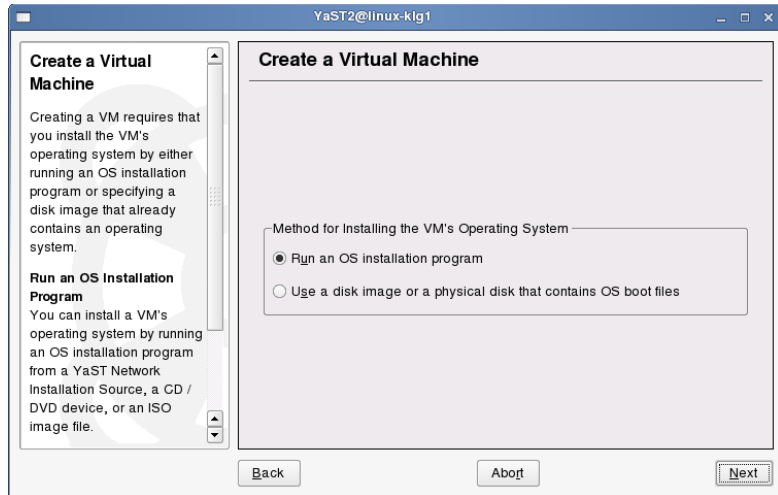
In our example there is already one guest domain running on the system, which is listed in the upper part of the dialog. The columns of the table display various information about the domain including the name, the status and the memory allocation.

The following buttons are in the lower part of the dialog:

- **Add.** Select this button to create a new domain.
- **Refresh.** This button refreshes the information about the domains.
- **Delete.** Deletes a domain completely.
- **Start.** Starts a domain.
- **View.** Opens a terminal window to access the console of a domain.
- **Shutdown.** Performs a regular shutdown of the guest OS.
- **Terminate.** Terminates the domain immediately without waiting for the guest OS to shutdown.

After selecting **Add**, the following dialog appears:

**Figure 8-6**

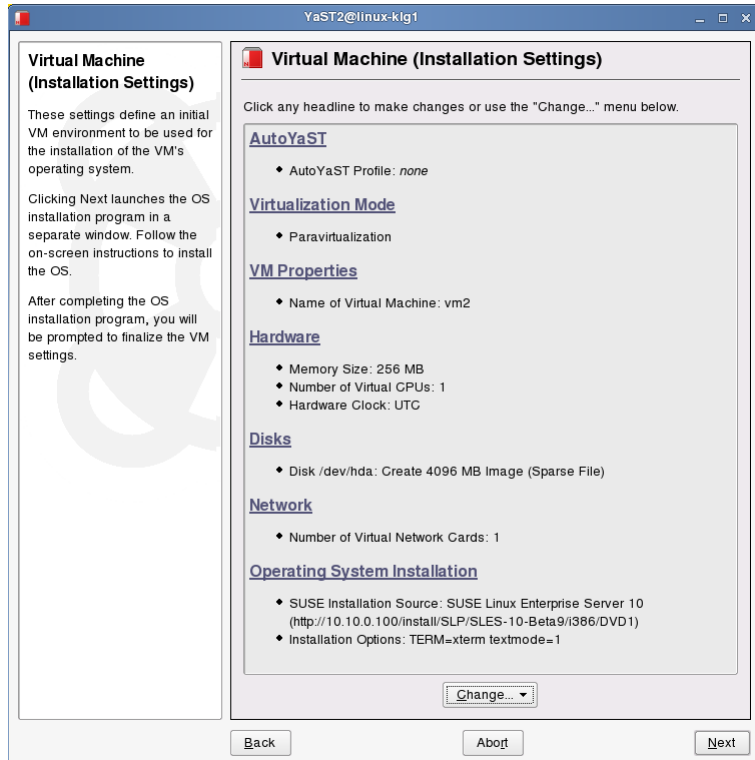


The dialog gives you two choices:

- **Run an OS installation program.** This allows you to run a SUSE Linux Enterprise Server installation from an installation source that is registered in the system.
- **Use a disk image or a physical disk that contains OS boot files.** This option lets you create a Xen domain from an existing installation in a physical disc or disc image.

For the following example we select the **Run an OS installation program** option. The following dialog appears:

**Figure 8-7**

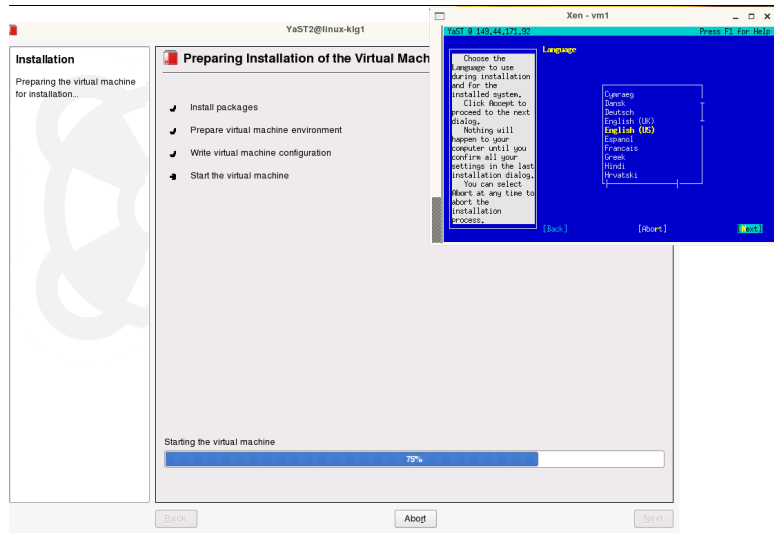


The following options can be adjusted by selecting their headlines:

- **AutoYaST.** In this option you can specify an AutoYaST profile that should be used for the installation. When there is no AutoYaST profile, a manual installation is started.
- **Virtualization.** You can switch between para virtualization and full virtualization. Full-virtualization is only available on supported hardware with Intel or AMD virtualization extension.
- **VM Properties.** Here you can change the name of the new domain.
- **Hardware.** In this option you can configure the hardware configuration of the domain. (Memory, Number of CPUs, ...)
- **Disks.** Configure the Disks here. These can either be physical block devices or file system / disc images.
- **Network.** This option lets you add additional network adapters to the domain.
- **Operating System Installation.** Here you can configure the installation source and additional installation options.

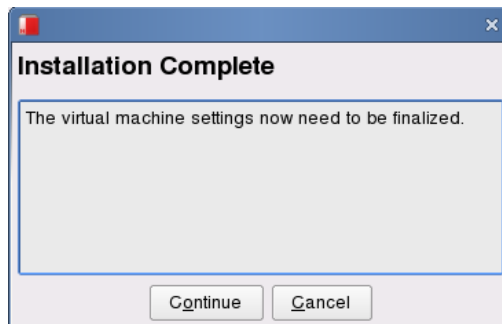
For our example we stay with the default and select **Next**. Now the domain environment and the installation process is started.

**Figure 8-8**



The installation itself is a standard SUSE Linux Enterprise Server installation, except that it runs in text mode. After the packages have been installed, the following dialog appears:

**Figure 8-9**





Select **Continue**.

The following dialog gives you a resume about the domain configuration. Usually there is nothing to do here. Select **Next** in this dialog and in the domain overview.

A terminal window opens up where you can finish the remaining steps of the OS installation with YaST.

## **Exercise 8-2      *Install a Guest Domain***

In this exercise, you learn how to install a Xen guest domain using YaST. Before you start with this exercise ,you must have installed xen on your system.

Do the following:

1. Open the **YaST Control Center**.
2. Select **System > Virtual Machine Management**.
3. Select **Add**.
4. Select **Run an OS installation program** and then **Next**.
5. Select **Next**.
6. After a while, a terminal window opens and a standard SUSE Linux Enterprise Server installation starts up. Select this window.
7. Press **Alt+N**.
8. Use the tab-key to navigate to the item “**Yes, I Agree to the License Agreement**”. Then press the space bar.
9. Press **Alt+N**.
10. Press **Alt+N**.
11. (Optional) Adjust the settings for **Region** and **Time Zone**.  
Navigate to the menus with the tab-key and use the arrow keys to change an option.
12. Press **Alt+N**.
13. Confirm the installation overview by pressing **Alt+A**.
14. Start the installation by pressing **Alt+I**.
15. (Wait till the installation has been finished.)
16. Select **Continue** in the **Installation Complete** message box.
17. Select **Next** in the domain configuration overview.

18. Select **Finish** in the **Virtual Machine Started** message box.
19. Switch to the terminal of the virtual domain.
20. Select **Next** (Press **Alt+P**).
21. Enter **novell** as root password. Select **Next** to continue (Press **Alt+N**).
22. Accept that the password is too simple.
23. Select **Alt-n** to continue.
24. Select **No, Skip this Test** (Press **Alt+O**).
25. Select **Next** (**Alt+N**).
26. Select **Next** (**Alt+N**).
27. Select **Next** (**Alt+N**).
28. Create user **geeko** with the password **novell**.
29. Select **Next** (**Alt+N**).
30. Accept that the password is too simple.
31. Select **Next** (**Alt+N**).
32. Select **Next** (**Alt+N**).
33. Select **Finish** (**Alt+F**).
34. Test if you can login to the new domain as the user root with the password novell.
35. Please do not close the terminal window, we will use it in the next exercise.

***(End of Exercise)***

## Objective 5    **Manage Xen Domains at the Command Line**

In the following you learn how to manage Xen domains at the command line. This includes:

- Understand a Domain Configuration File
- Use the xm Tool
- Automate Domain Startup and Shutdown

### ***Understand a Domain Configuration File***

Every Xen domain needs a configuration file. For domains which have been created with YaST, the configuration file is usually located in **/etc/xen/vm/**.

Under **/etc/xen/examples**, you find two example files, which can be used if you would like to create a configuration from scratch.

- **xmexample1**. This is a template configuration file for a single domain.
- **xmexample2**. This is an example for multiple domain configurations in one file.

For the beginning, **xmexample1** is a better choice.

A configuration file contains several keywords which configure different aspects of a Xen domain. The following is an example configuration file using the most common options. The # character is used for comments. Please read the comments in the example for details about the configuration options.

```
# Unique name of the domain
name = "SLES10-WebServer"

# The following lines point to the kernel and initrd file
# on the filesystem of the domain. The filesystem itself is
# defined later.
kernel = "/boot/vmlinuz-Xen"
ramdisk = "/boot/initrd-Xen"

# The amount of memory that is initially assigned to the
# domain. This can be changed at runtime.
memory = 256

# The next line defines a some details about the network
# configuration. When left blank, defaults are used,
# which work fine in most cases.
vif = [ ' ' ]

# This defines the disc of the domain. "phy" means that the
# physical device /dev/hda1 is mapped to the virtual device
# /dev/hda1 in the domain. "w" indicates, that the disc is
# writable.
disk = [ 'phy:hda1,hda1,w' ]

# The following is an example for a file based filesystem
# image. In this case the "file:" keyword is used.
# disk = [ 'file:/data/vm/SLES10-disc.img,hda1,w' ]

# Sets the device for the Linux Kernel
root = "/dev/hda1 ro"
```



A good source for detailed documentation and howtos about Xen and the domain configuration files is the Xen wiki at:  
<http://wiki.xensource.com/xenwiki/>

---

## *Use the xm Tool*

**xm** is the administration tool for Xen domains. **xm** communicates with the **xend** management process running on the domain0 Linux installation.

The following is the general format of a **xm** command line:

```
xm command [options] [arguments] [variables]
```

You can get a complete list of the most common **xm** commands by entering **xm help**. A complete list can be viewed with **xm help --long**. It is also possible to display specific information about a certain command with **xm help [command\_name]**.

To start a virtual machine, the **create** command is used:

```
xm create -c -f /data/xen/SLES10-WebServer.conf
```

The **-c** option lets **xm** connect to the terminal of the started domain, so that you can interact with the system. To disconnect from the terminal and return to the original command line, enter the key combination **Ctrl-]**.

The **-f** option specifies the configuration file of the domain that should be started.

The command **list** displays information about the currently running Xen domains:

```
xm list
```

The output of the **list** command contains the following fields:

- **name.** The name of the domain as specified in the configuration file.

- **domid.** A numeric, consecutive domain ID, which is automatically assigned when the domain starts.
- **memory.** The amount of memory assigned to the domain.
- **vcpus.** The number of virtual CPUs utilized by this domain.
- **state.** The current state of the domain. This could be:
  - **r.** The domain is running.
  - **b.** The domain has been created, but is currently blocked. This can happen, when a domain is waiting for I/O or when there is nothing to do for domain.
  - **p.** The domain is paused. The state of the domain is saved and can be restored.
  - **s.** The domain is in the process of being shutdown.
  - **c.** The domain is crashed, due to an error or missconfiguration.

An alternative to list is the command **top**, which displays domain information updated in realtime.

The **console** command connects you with the terminal of a running domain:

```
xm console <domain_id>
```

The command takes the domain id as a parameter, which can be determined with the **list** command (field domid). As mentioned before, use the key combination **Ctrl-]** to disconnect from a terminal.

With the **pause** command you can interrupt the execution of a domain temporarily:

```
xm pause <domain_id>
```

A paused domain is not completely shut down. The current state is saved and the execution of the domain can be continued with the **unpause** command:

```
xm unpause <domain_id>
```

To shutdown a domain, use the **shutdown** command:

```
xm shutdown <domain_id>
```

In case the domain is not responding anymore, you can force the shutdown of the domain with the **destroy** command:

```
xm destroy <domain_id>
```

To save the state of a domain for a longer time (eg. over a reboot of domain0) you can use the **save** command:

```
xm save <domain_id> <filename>
```

The domain can be restored from the resulting file with the **restore** command:

```
xm restore <filename>
```

Another commonly used command is **mem-set**, which allows you to change the memory allocation of a domain:

```
xm mem_set <domain_id> <amount_of_memory>
```

The amount of memory is specified in megabytes.





---

Instead of the domain ID `<domain_id>`, you can also use the domain name in all `xm` commands.

---

### **Exercise 8-3      *Change Memory Allocation of a Guest Domain***

In this exercise, you learn how to change the memory allocation of a guest domain by changing the domain configuration file.

The following assumes, that you still have an open terminal window of the guest domain, that you have configured in the previous exercises.

Do the following:

1. Open a terminal window and **su -** to the root user.
2. Enter the command **xm list**.
3. Note the memory allocation of the domain **vm1**.
4. Switch to the terminal of the Xen domain and halt the system by typing **halt**. Wait till the system has been halted.
5. Return to the root terminal and use the command **xm list** to verify that the domain **vm1** is not running anymore.
6. Open the file **/etc/xen/vm/vm1** with a text editor.
7. Look for the **memory** parameter and change the value to **172**.
8. Save and close the file.
9. Enter the following command to start the domain:

```
xm create -c -f /etc/xen/vm/vm1
```

10. Wait till the system has been booted and you see the login prompt.
11. Press the key combination **Ctrl-]** to detach from the domain terminal and return to the root terminal.
12. Use the command **xm list** to determine the memory allocation of domain **vm1**. It should be 172MB.
13. Also note the **ID** of domain **vm1**.

14. Attach to the terminal of vm1 with the command  
**xm console <noted\_id>**

***(End of Exercise)***

## ***Automate Domain Startup and Shutdown***

When you start, shutdown or reboot domain0 of a Xen system, this also affects the other running Xen domains. Without a running domain0, the other Xen domains cannot operate.

SUSE Linux Enterprise Server 10 comes with a start script called **xendomains** which is included in the package **xen-tools**.

The script should be installed on domain0 and does the following:

- When domain0 is booted, all domains with configuration files located under **/etc/xen/auto/** are started.
- When domain0 is shutdown or rebooted, running Xen domains are shutdown automatically.

The script has some configuration options, which can be adjusted in the file **/etc/sysconfig/xendomains**. The configuration variables in this file are well documented.

One interesting option is to migrate domains automatically to a different host when a domain0 is shutdown. This can be configured in the variable **XENDOMAINS\_MIGRATE**. The variable has to be set to the IP address of the target machine. When the variable is empty, no migration is performed.

## **Exercise 8-4      Automate Domain Startup**

In this exercise, you learn how to startup domains automatically when the system is booted.

Do the following:

1. Open a terminal window and **su -** to the root user.
2. Move the vm1 configuration file into the auto directory:  
**mv /etc/xen/vm/vm1 /etc/xen/auto/**
3. Shutdown vm1 with the command **xm shutdown vm1**.
4. Wait a moment and control with the command **xm list** if the domain has been shut down. Continue with next step when the domain vm1 is not listed anymore.
5. Reboot you system by entering **reboot**.
6. At the boot prompt, select the **Xen** entry.
7. When the system has been started up, log in to the graphical interface as user geeko with the password novell.
8. Open a terminal window and **su -** to the root user.
9. Enter the command **xm list**.
10. The domain vm1 should have been automatically started and should be listed in the xm output.

***(End of Exercise)***

## Objective 6      Understand Xen Networking

Usually the network connection of Xen domains works out of the box. However, in case you would like to change the configuration, networking with Xen can be a bit tricky. The following should give you an overview of how Xen domains are connected to the physical network.

To better understand the concept of Xen networking, do the following:

- Understand the Basic Networking Concept
- Understand Bridging
- Understand the Network Interfaces in domain0

### ***Understand the Basic Networking Concept***

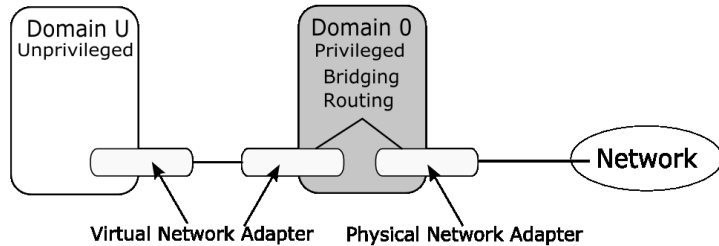
In a Xen setup, domain0 is controlling the physical network interfaces of a host system. Unprivileged domains are connected to domain0 through virtual ethernet adapters.

One virtual adapter in an unprivileged domain is connected to one virtual adapter in domain0.

In domain0, standard Linux networking mechanisms like bridging or routing are used to connect the virtual adapters through the physical adapter to the network.

The following is an illustration of this basic concept:

**Figure 8-10**



### ***Understand Bridging***

On SUSE Linux Enterprise Server 10, the default mechanism to connect virtual and physical interfaces in domain0 is bridging. Other mechanisms like routing with or without Network Address Translation (NAT) are out of the scope of this course.

Bridging basically means that multiple network interfaces are combined to one. Traditionally this technique is used to connect two physical network interfaces or network segments.

In a Xen system, bridging is used to connect virtual and physical network adapters in domain0. In a Xen system, you can consider the bridge as a kind of virtual switch which all virtual and physical interfaces are connected to.

The configuration of the bridge is done by the xend management process. When a new domain is created, the following changes to the network configuration are made (simplified):

1. Xen provides a virtual interface to the new domain.
2. xend creates a new virtual interface in domain0.

3. Both virtual interfaces are connected through a virtual point to point connection.
4. The virtual interface in domain0 is added to the bridge with the physical interface.

These steps only affect the general network connectivity. The IP configuration in the Xen domains has to be done separately with DHCP or a static network configuration.

xend is performing these network changes with the help of scripts, which are located at **/etc/xen/scripts/**. The following scripts are used for bridged networking:

- **network-bridge**. This script is called initially when xend is started. It sets up the bridge **xenbr0** and moves the physical interfaces onto that bridge.
- **vif-bridge**. This script is called for every domain that is started and adds the virtual interface to the bridge.

In the file **/etc/xen/xend-config.sxp** you can configure which network scripts are used by xend.

### ***Understand the Network Interfaces in domain0***

When you look at the network interfaces in domain0 with the command **ip a**, you can see that there are many more interfaces than in a regular Linux installation.



The following is an example output of **ip a** on domain0 (shortened):

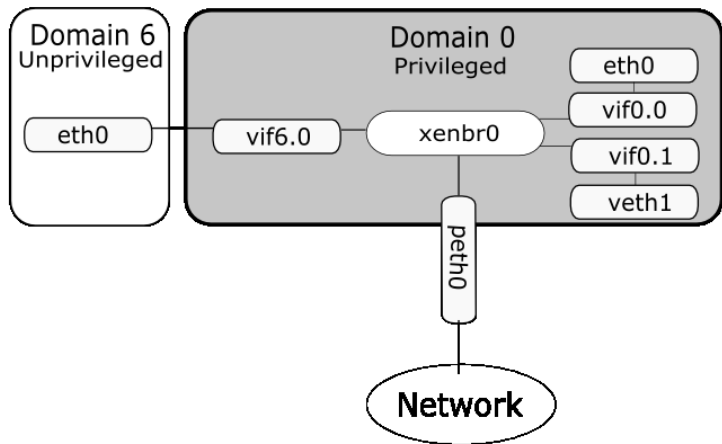
```
linux-3rsm:~ # ip a
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: peth0: <BROADCAST,MULTICAST,NOARP,UP> mtu 1500 qdisc
pfifo_fast qlen 100
    link/ether fe:ff:ff:ff:ff:ff brd ff:ff:ff:ff:ff:ff
    inet6 fe80::fcff:ffff:feff:ffff/64 scope link
        valid_lft forever preferred_lft forever
4: vif0.0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue
    link/ether fe:ff:ff:ff:ff:ff brd ff:ff:ff:ff:ff:ff
    inet6 fe80::fcff:ffff:feff:ffff/64 scope link
        valid_lft forever preferred_lft forever
5: eth0: <BROADCAST,MULTICAST,NOTRAILERS,UP> mtu 1500 qdisc
noqueue
    link/ether 00:11:25:81:4c:5b brd ff:ff:ff:ff:ff:ff
    inet 149.44.171.67/23 brd 149.44.171.255 scope global
eth0
    inet6 2001:780:101:aa00:211:25ff:fe81:4c5b/64 scope
global dynamic
        valid_lft 29998sec preferred_lft 9996sec
    inet6 fe80::211:25ff:fe81:4c5b/64 scope link
        valid_lft forever preferred_lft forever
6: vif0.1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop
    link/ether fe:ff:ff:ff:ff:ff brd ff:ff:ff:ff:ff:ff
7: veth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
[...]
19: veth7: <BROADCAST,MULTICAST> mtu 1500 qdisc noop
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
20: xenbr0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue
    link/ether fe:ff:ff:ff:ff:ff brd ff:ff:ff:ff:ff:ff
    inet6 2001:780:101:aa00:fcff:ffff:feff:ffff/64 scope
global dynamic
        valid_lft 29998sec preferred_lft 9996sec
    inet6 fe80::200:ff:fe00:0/64 scope link
        valid_lft forever preferred_lft forever
23: vif3.0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue
    link/ether fe:ff:ff:ff:ff:ff brd ff:ff:ff:ff:ff:ff
    inet6 fe80::fcff:ffff:feff:ffff/64 scope link
        valid_lft forever preferred_lft forever
```

The following interface naming schema is used in domain0:

- **peth**. These are **physical** interfaces in domain0. peth devices are connected to the network bridge.
- **vif**. These are **virtual interfaces** which are part of the bridge. The name of a vif interface identifies to which domain this interface is connected to. **For example:** vif6.0 is connected to the first virtual interface in domain 6.
- **veth**. These **virtual interfaces** are connected to the vif interfaces of domain0 (vif0.x). By default 7 vif <-> veth pairs are created. The veth interfaces can be used for more complex network setups.
- **eth0**. The first veth interface is named eth0 and connected with vif0.0. This is the “default” network interface of domain0.
- **xenbr0**. This is the default bridge that connects virtual and physical interfaces.

The following illustration gives you an overview of the interfaces in domain0.

**Figure 8-11**



You can use the command **brctl show** in domain0, to see which interfaces have been added to the network bridge.



---

Due to the complexity of the Xen network setup, the default firewall (SuSEFirewall2) is not working correctly in domain0. It is therefore recommended to disable SuSEFirewall2 and to setup a customized firewall if needed.

---

### **Exercise 8-5      Check the Network Configuration**

This exercise assumes that you have a Xen system with domain 0 and one more Xen domain running.

Do the following:

1. Open a terminal window and **su -** to the root user.
2. Make sure that the domain **vm1** is running by typing the command **xm list**.
3. In the output of the **xm** command, note the **ID** of the domain **vm1**.
4. View the network bridge configuration with the command **brctl show**.
5. You should see the configuration of the bridge **xenbr0**. The interfaces **peth0** (physical interface) **vif0.0** (virtual interface of domain 0) and the virtual interface **vifx.0** (where **x** is the domain ID of domain **vm1**) should be added to the bridge.
6. Shutdown the domain with the command **xm shutdown vm1**.
7. Wait a moment and control with the command **xm list** if the domain has been shut down. Continue with next step when the domain **vm1** is not listed anymore.
8. Enter the command **brctl show** again. Note that the interface of the domain **vm1** has been removed from the bridge.
9. Restart the domain with: **xm create -f /etc/xen/vm/vm1**
10. Note the ID of **vm1** and check with **brctl show** if the interface of **vm1** has been added again.

**(End of Exercise)**

## Objective 7      **Migrate a Guest Domain**

One advantage of virtualization is that domains can easily be moved from one physical system to another. Under Xen this procedure is called a **domain migration**.

A domain migration is performed by copying the current memory content. Please note the following before migrating a domain:

- There is no automatic way to copy the mass storage devices of a domain to another system. You have to make sure that the file systems (file system images or physical partitions) are available on the current and on the new host system. This can either be done by copying the data manually or by using a distributed file system (like NFS or SAN/NAS storage solutions).
- When a domain is migrated, the network settings are not automatically adjusted. Therefore the current and the new host system have to be in the same subnet or the network settings have to be manually adjusted after the migration.

You have the following two options to migrate a Xen domain:

- Use Domain Save and Restore
- Use Migration and Live Migration

### ***Use Domain Save and Restore***

A very simple way to migrate a domain is to use the save and restore function of the `xm` tool.

With the command **`xm save <domain_id> <filename>`**, you can suspend the specified domain and save the status to the given filename.

This file can then be copied to the new host system. To restore the domain, use the command **`xm restore <filename>`**.

As mentioned above, besides the file created with `xm`, you might also have to copy the filesystems to the new host system.

### ***Use Migration and Live Migration***

Instead of the `save` and `restore` commands of `xm`, you can also use the command **`xm migrate <domain_id> <target_host>`**. This command migrates a domain directly to a new host. In this case it's not necessary to copy memory state files manually.

In order to get this working, the current and new host must be running Xen and `xend`.

By adding the option **`--live`** to the migration command line, the downtime during the migration can be reduced to typically 60-300ms. Instead of shutting down the domain before the migration starts, Xen attempts to keep it running while the migration is in progress.

The `xend` configuration file **`/etc/xen/xend-config.sxp`** contains two options concerning domain migration:

**`(xend-relocation-server yes)`**

This option enables the migrating functionality in `xend`.

**`(xend-relocation-hosts-allow '^localhost$')`**

This option controls which hosts are allowed to connect to `xend` for domain migration. By default, only `localhost` is allowed to connect. The option takes regular expressions as parameter. Have a look at the configuration file for examples.



Please note, that there are two `xend` involved in a domain migration (current and new host). You might have to adjust the `xend-config.sxp` file on both systems.

---

## Summary

Objective	Summary
1. Understand the Concept of Virtualization	
2. Understand How Xen Works	<p>There are two different kinds of virtualization:</p> <ul style="list-style-type: none"><li>■ Full-Virtualization</li><li>■ Para-Virtualization</li></ul> <p>Xen uses para-virtualization. It provides access to the physical hardware through an API.</p>
3. Install Xen	<p>The following packages have to be installed in the initial SUSE Linux Enterprise Server 10 installation:</p> <ul style="list-style-type: none"><li>■ <b>xen</b>. This package contains the Xen Virtual Machine Monitor (Hypervisor).</li><li>■ <b>xen-tools</b>. Contains xend and a collection of command line tools to administer a Xen system.</li><li>■ <b>kernel-xen</b>. This package contains a modified Linux kernel that runs in a Xen domain.</li><li>■ <b>xen-doc-*</b> (optional). Xen documentation in various formats.</li></ul> <p>The installation of xen adds an entry in the grub configuration file.</p>

Objective	Summary
4. Manage Xen Domains with YaST	<p>YaST provides a module which can be used to create and manage Xen domains. The module is called: <b>Virtual Machine Management (Xen)</b>.</p> <p>This module offers a convenient way to create and control the Xen domains on your system. The module can be started from the <b>System</b> section in the YaST Control Center, and has to run on the Linux system running in domain0.</p>
5. Manage Xen Domains at the Command Line	<p>Every Xen domain needs a configuration file. Usually this is located in <b>/etc/xen/vm/</b>.</p> <p><b>xm</b> is the central administration tool for xen domains.</p> <p>To start a virtual machine, the <b>create</b> command is used. For example:</p> <pre>xm create -c -f SLES10.conf</pre> <p>Some services are not required in a xen environment and can be removed.</p> <ul style="list-style-type: none"><li>■ <b>insserv -r earlykbd</b></li><li>■ <b>insserv -r kbd</b></li><li>■ <b>insserv -r irq_balancer</b></li></ul> <p>Under Xen, all domains are connected with the physical network through domain0.</p>



Objective	Summary
6. Understand Xen Networking	<p>Domain0 is the central point to configure the network connections on a Xen system.</p> <p>A network bridge in domain0 is used as a virtual switch.</p> <p>This bridge is set up and controlled by xend.</p>
7. Migrate a Guest Domain	<p>One advantage of virtualization is, that domains can easily be moved from one physical system to another. Under Xen this procedure is called a <b>domain migration</b>.</p> <p>Domains can be migrated with xm's <b>save</b> and <b>restore</b> commands or with the <b>migrate</b> command.</p>



## SECTION 9    Configure a DNS Server Using BIND

This section contains some changes concerning the commands `dnssec-keygen` and `nsupdate`.

### Objectives

1. Create a Key for Zone Transfer
2. Configure Dynamic DNS

## Objective 1      Create a Key for Zone Transfer

To create a key, use the **dnssec-keygen** command. The file name of the key is printed on the screen:

```
da51:/var/lib/named # dnssec-keygen -a HMAC-MD5 -b 128 -n HOST
zonetransfer
Kzonetransfer.+157+01389
```

The options are explained in the following table:

**Table 9-1**

Option	Description
-a HMAC-MD5	The encryption procedure used (here HMAC-MD5)
-b 128	The length of the key (in the example, 128 bits)
-n HOST	The type of key
zonetransfer	Name of key

If you use this command, two files are created in the current directory:

```
da51:/var/lib/named # ls -l K*
-rw----- 1 root root 56 Feb 21 10:39 Kzonetransfer.+157+01389.key
-rw----- 1 root root 81 Feb 21 10:39 Kzonetransfer.+157+01389.private
```

- The \*.key file contains a DNS KEY record that can be included in a zone file using the **include** statement.
- The \*.private file contains algorithm specific fields. For security reasons, this file does not have general read permission.



---

The numbers at the end of the filename will be slightly different when you run this command.

---

Both files contain the same key:

```
da51:/var/lib/named # cat Kzonetransfer.+157+01389.key
zonetransfer. IN KEY 512 3 157 KhDVspogFonWKv58rFXOWw==
da51:/var/lib/named # cat Kzonetransfer.+157+01389.private
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key: KhDVspogFonWKv58rFXOWw==
```

This key has to be included in the configuration file `/etc/named.conf` on both the master server and the slave server.

## Objective 2      Configure Dynamic DNS

If the number of zones and hosts increases, it is inconvenient to edit the zone files manually.

You can modify the resource record sources of the name server dynamically without editing and reloading files. This is called dynamic DNS. Dynamic DNS can also be used by external services like DHCP.

To allow dynamic changes, add the following line in the zone definition.

```
allow-update { 127.0.0.1; };
```

In this example, only the loopback address is used, but it is also possible to add IP addresses of other hosts.

To edit the DNS records dynamically, use the command **nsupdate**:

```
da51:~ # nsupdate
>
```



The security tool AppArmor is installed and enabled on SUSE Linux Enterprise Server 10 by default. There is an AppArmor profile for BIND that prohibits **nsupdate** to change the BIND zone information.

AppArmor is covered in the course *SUSE Linux Enterprise Server 10 Security* (Course 3075) in detail.

Stop the AppArmor daemon by entering **rcapparmor stop** to use **nsupdate** as described in the following.

---



Before using **nsupdate**, make sure that all zone files have the correct access permissions. If they are not set properly, use the following commands:

```
chgrp -R named /var/lib/named  
chmod -R g+w /var/lib/named
```

---

Dynamic updates are stored in a journal file for the zone. This file is automatically generated by the server when the first dynamic update is performed. The name of the journal file is created by appending the extension `.jnl` to the name of the corresponding zone file. The journal file is in a binary format and should not be edited manually.

The contents of the journal file are written to the zone file every 15 minutes. When the name server is shut down, the contents of the journal file are written to the zone file, too.



---

Dynamic updates will change the layout of your zone files when the data is written to them. You should either use an editor or **nsupdate** to modify your zone information instead of using both tools.

---

The most important **nsupdate** options are

- **server *server* [*port*]**. Sends all dynamic update requests to the name server *server*.

If no *port* number is specified, the default DNS port number 53 is used.

- **update delete *reference* [*tll*] [*class*] [*type* [*value...*]]**. Deletes any resource records named *reference*.

If the record type and value are provided, only matching resource records will be removed.

The Internet class (IN) is assumed if *class* is not supplied.

*tll* is ignored. It is only allowed for compatibility.

- **update add *reference tll* [*class*] *type value...***. Adds a new resource record with the specified *tll* (in seconds), *class* and *value*.
- **send**. Sends the current message. This is necessary to actually execute the command. This is equivalent to entering a blank line.



---

All commands are described in the man page of `nsupdate`: **man 8 nsupdate**.

---

The following is an example of using **nsupdate**:

```
da51:~ # nsupdate
> server 127.0.0.1
> update add dal3.digitalairlines.com 86400 A 10.0.0.13
>
> update delete dal3.digitalairlines.com A
>
> update add 13.0.0.10.in-addr.arpa 86400 PTR dal3.digitalairlines.com
>
```

This will generate messages like the following in `/var/log/messages`:

```
May 20 11:13:58 da51 named[5161]: client 127.0.0.1#32781: updating zone
'digitalairlines.com/IN': adding an RR
May 20 11:13:58 da51 named[5161]: journal file
master/digitalairlines.com.zone.jnl does not exist, creating it
May 20 11:13:58 da51 named[5161]: zone digitalairlines.com/IN: sending
notifies (serial 2005051903)
May 20 11:21:50 da51 named[5161]: client 127.0.0.1#32783: updating zone
'0.0.10.in-addr.arpa/IN': adding an RR
May 20 11:21:50 da51 named[5161]: journal file master/10.0.0.zone.jnl does
not exist, creating it
May 20 11:21:50 da51 named[5161]: zone 0.0.10.in-addr.arpa/IN: sending
notifies (serial 2005051902)
```

The journal files will be created automatically:

```
da51:/var/lib/named # dir master/
total 16
drwxrwxr-x  2 root  named 200 May 20 11:21 .
drwxrwxr-x  9 root  named 408 May 20 10:40 ..
-rw-rw-r--  1 root  named 463 May 19 12:06 10.0.0.zone
-rw-r--r--  1 named named 814 May 20 11:21 10.0.0.zone.jnl
-rw-rw-r--  1 root  named 440 May 19 14:51 digitalairlines.com.zone
-rw-r--r--  1 named named 794 May 20 11:13 digitalairlines.com.zone.jnl
```

Press **Ctrl + D** or enter **quit** to quit `nsupdate`.



In the following table, the most important record types are listed:

**Table 9-2**

Record Type	Meaning	Value
SOA	Start of Authority	Parameter for the domain
NS	DNS server	Name of a DNS server for this domain
MX	Mail exchanger	Name and priority of a mail server for this domain
A	Address	IP address of the computer
PTR	Pointer	Name of the computer
CNAME	Canonical name	Alias name for the computer

Alternatively, you can specify a file containing the needed commands. An ASCII file (called **updates**) with the following content delivers the same result as the interactive DNS update as shown above:

```
da51:~ # cat updates
update delete da7.digitalairlines.com
update add da8.digitalairlines.com 84600 A 10.0.0.8
da51:~ # nsupdate -v -k Kdhcp-dns.+157+23165.key updates
da51:~ #
```

## Summary

Objective	Summary
1. Create a Key for Zone Transfer	<p>Generate the key by using the <b>dnssec-keygen</b> command.</p> <p>The *.key file contains a DNS KEY record that can be included in a zone file using the include statement.</p> <p>The *.private file contains algorithm specific fields. For security reasons, this file does not have general read permission.</p> <p>This key has to be included in the configuration file /etc/named.conf on both the master server and the slave server.</p>
2. Configure Dynamic DNS	<p>You can to modify the resource records of BIND dynamically and without editing and reloading files.</p> <p>Dynamic DNS can also be used by external services like DHCP.</p> <p>To manipulate the DNS records dynamically, use the <b>nsupdate</b> command.</p>

## SECTION 10 Configure DHCP Pools and Failover

DHCP pools and failover are not new features of the dhcpd. But they are new in the CLE curriculum.

### Objectives

1. Configure DHCP Pools
2. Configure DHCP Failover

## Objective 1    Configure DHCP Pools

The **pool** declaration can be used to specify a pool of addresses that will be treated differently than any other pool of addresses, even on the same network segment or subnet.

For example, you may want to provide a large set of addresses that can be assigned to DHCP clients that are registered to your DHCP server, while providing a smaller set of addresses, possibly with short lease times, that are available for unknown clients.

To do this, you would set up a pair of **pool** declarations:

```
subnet 10.0.0.0 netmask 255.255.255.0 {
    option routers 10.0.0.254;

    # Unknown clients get this pool.
    pool {
        option domain-name-servers 10.0.0.254;
        max-lease-time 300;
        range 10.0.0.100 10.0.0.250;
        allow unknown-clients;
    }

    # Known clients get this pool.
    pool {
        option domain-name-servers 10.0.0.251, 10.0.0.252;
        max-lease-time 28800;
        range 10.0.0.5 10.0.0.99;
        deny unknown-clients;
    }
}
```

It is also possible to set up entirely different subnets for known and unknown clients. “Known clients” mean, that an IP address is assigned to a MAC address using the **host** statement. Pools exist at the level of shared networks, so address ranges within pool declarations can be on different subnets.

As you can see in the preceding example, pools can have permit lists that control which clients are allowed access to the pool and which aren't.

Each entry in a pool's permit list is introduced with the allow or deny keyword. If a pool has a permit list, then only those clients that match specific entries on the permit list will be eligible to be assigned addresses from the pool.

If a pool has a deny list, then only those clients that do not match any entries on the deny list will be eligible. If both permit and deny lists exist for a pool, then only clients that match the permit list and do not match the deny list will be allowed access.

## Objective 2    **Configure DHCP Failover**

This objective covers the following topics:

- Basics of DHCP Failover
- Configure Failover

### ***Basics of DHCP Failover***

The failover protocol allows two DHCP servers (and no more than two) to share a common address pool. Each server will have about half of the available IP addresses in the pool at any given time for allocation.

If one server fails, the other server will continue to renew leases out of the pool, and will allocate new addresses out of the half of available addresses that it had before communication with the other server was lost.

When a server starts that has not previously communicated with its failover peer, it must establish communications and synchronize with the peer before it can serve clients. This can happen either because you have just configured your DHCP servers to perform failover for the first time, or because one of your failover servers has failed and lost its database.

The initial recovery process is designed to ensure that when one failover peer loses its database and then resynchronizes, any leases that the failed server gave out before it failed will be honored. When the failed server starts up, it notices that it has no saved failover state and attempts to contact its peer.

When it has established contact, it asks the peer for a complete copy of its lease database. The peer then sends its complete database and sends a message indicating that it is done. The failed server then waits until MCLT (*Maximum Client Lead Time*) has passed. Once MCLT has passed, both servers make the transition back into normal operation.

This waiting period ensures that any leases the failed server may have given out while out of contact with its partner will have expired.

While the failed server is recovering, its partner remains in the partner-down state, which means that it is serving all clients. The failed server provides no service at all to DHCP clients until it has made the transition into normal operation.

In the case where both servers detect that they have never before communicated with their partner, they both come up in this recovery state and follow the procedure we have just described. In this case, no service will be provided to DHCP clients until MCLT has expired.

## ***Configure Failover***

In a failover configuration you have to decide which server acts as the primary and which acts as the secondary server. Both servers need to have the same configuration for the basic DHCP service they are sharing. They only differ in the failover setup. Therefore, it is recommended to have a common configuration file on both machines which is included in the file `/etc/dhcpd.conf`.

The configuration file for the primary server looks like this:

```
failover peer "digitalairlines" {
    primary;
    address 10.0.0.10;
    port 847;
    peer address 10.0.0.12;
    peer port 647;
    max-response-delay 180;
    mclt 1800;
    split 128;
    load balance max seconds 3;
}

# Now we include the identical configuration on both
# machines
include "/etc/dhcpd.conf.master";
```

The statements in this example peer declaration are as follows:

- **primary** and **secondary**. Determines whether the server is primary or secondary, as described earlier.
- **address**. Specifies the IP address or DNS name on which the server should listen for connections from its failover peer.
- **port**. Specifies the TCP port on which the server should listen for connections from its failover peer (default: 647 and 847). This parameter must be specified, because the failover protocol does not yet have a reserved TCP port number.
- **peer address**. Specifies the IP address or DNS name to which the server should connect to reach its failover peer.
- **peer port**. Specifies the TCP port to which the server should connect to reach its failover peer for failover messages. This parameter must be specified, because the failover protocol does not yet have a reserved TCP port number. The **peer port** can be the same as the **port**.
- **max-response-delay**. Specifies how many seconds the DHCP server waits without receiving a message from its failover peer before it assumes that connection has failed.



This number should be small enough that a transient network failure breaks the connection will not result in the servers being out of communication for a long time, but large enough that the server isn't constantly making and breaking connections.

This parameter must be specified.

- **mclt.** Defines the *Maximum Client Lead Time*. It must be specified on the primary, and can be specified also on the secondary server. This is the length of time for which a lease may be renewed by either failover peer without contacting the other.

The longer you set this, the longer it will take for the running server to recover IP addresses after moving into PARTNER-DOWN state. The shorter you set it, the more load your servers will experience when they are not communicating.

A value of 1800 is recommended.

- **split.** Specifies the split between the primary and secondary server for the purposes of load balancing.

Whenever a client makes a DHCP request, the DHCP server runs a hash on the client identification. If the hash comes out to less than the split value, the primary answers. If it comes out to equal to or more than the split, the secondary answers.

A meaningful value is 128 and can only be configured on the primary server.

- **load balance max seconds.** Specifies a cutoff after which load balancing is disabled. The cutoff is based on the number of seconds since the client sent its first DHCPDISCOVER or DHCPREQUEST message.

The man pages recommend setting this to 3 or 5. The effect of this is that if one of the failover peers gets into a state where it is responding to failover messages but not responding to some client requests, the other failover peer will take over its client load automatically as the clients retry.

The configuration file for the secondary server looks like this:

```
failover peer "digitalairlines" {
    secondary;
    address 10.0.0.12;
    port 647;
    peer address 10.0.0.10;
    peer port 847;
    max-response-delay 180;
    load balance max seconds 3;
}

# Now we include the identical configuration on both
# machines
include "/etc/dhcpd.conf.master";
```

The differences to the primary server configuration are the statement “secondary”, the missing statements **mclt** and **split** and the interchanged values of **address**, **port**, **peer address** and **peer port**.

The main DHCP server configuration is contained in the file `/etc/dhcpd.conf.master`, which included in both configurations.



In order to find this file, it needs to be copied into the chroot environment of the DHCP server. The best way to achieve this is to modify the variable `DHCPD_CONF_INCLUDE_FILES` in `/etc/sysconfig/dhcpd`:  
`DHCPD_CONF_INCLUDE_FILES="/etc/dhcpd.conf.master"`

---

The master configuration file need to be modified:

```
ddns-update-style none;

default-lease-time 86400;
max-lease-time 86400;

option domain-name "digitalairlines.com";
option domain-name-servers 10.0.0.254;

option routers 10.0.0.254;

subnet 10.0.0.0 netmask 255.255.255.0 {
    pool {
        failover peer "digitalairlines";
        deny dynamic bootp clients;
        range 10.0.0.101 10.0.0.120;
    }
}
```

All failover configurations have to be defined in a **pool** statement. If you use several pools, you need to define the failover configuration in each of them.



In order to be aware of the name of the failover configuration, it has to be defined before the pool definition. That is why the include statement is written at the end of `/etc/dhcpd.conf`.

---

Failover is not supported on address allocation pools that contain addresses allocated to bootp clients. Therefore, the statement **deny dynamic bootp clients;** has to be defined.

When starting the DHCP server on the primary, you will see messages like these in /var/log/messages:

```
Jul  4 11:52:29 da10 dhcpd: failover peer digitalairlines:
I move from recover to startup
Jul  4 11:52:44 da10 dhcpd: failover peer digitalairlines:
I move from startup to recover
Jul  4 11:54:57 da10 dhcpd: failover peer digitalairlines:
peer moves from unknown-state to recover
Jul  4 11:54:57 da10 dhcpd: failover peer digitalairlines:
requesting full update from peer
Jul  4 11:54:57 da10 dhcpd: Sent update request all message
to digitalairlines
Jul  4 11:54:57 da10 dhcpd: failover peer digitalairlines:
peer moves from recover to recover
Jul  4 11:54:57 da10 dhcpd: failover peer digitalairlines:
requesting full update from peer
Jul  4 11:54:57 da10 dhcpd: Sent update done message to
digitalairlines
Jul  4 11:54:57 da10 dhcpd: Update request all from
digitalairlines: nothing pending
Jul  4 11:54:57 da10 dhcpd: failover peer digitalairlines:
peer update completed.
Jul  4 11:54:57 da10 dhcpd: failover peer digitalairlines:
I move from recover to recover-done
Jul  4 11:54:57 da10 dhcpd: failover peer digitalairlines:
I move from recover-done to normal
Jul  4 11:54:57 da10 dhcpd: failover peer digitalairlines:
peer moves from recover-done to normal
Jul  4 11:54:57 da10 dhcpd: pool 800e4138 10.0.0/24 total
20 free 20 backup 0 lts -10
Jul  4 11:54:57 da10 dhcpd: pool 800e4138 10.0.0/24 total
20 free 20 backup 0 lts 10
```

On the secondary server (which is started later), messages like these will appear in `/var/log/messages`:

```
Jul  4 11:55:52 da12 dhcpd: failover peer digitalairlines:  
I move from recover to startup  
Jul  4 11:55:52 da12 dhcpd: failover peer digitalairlines:  
peer moves from unknown-state to recover  
Jul  4 11:55:52 da12 dhcpd: failover peer digitalairlines:  
requesting full update from peer  
Jul  4 11:55:52 da12 dhcpd: failover peer digitalairlines:  
I move from startup to recover  
Jul  4 11:55:52 da12 dhcpd: Sent update request all message  
to digitalairlines  
Jul  4 11:55:52 da12 dhcpd: Sent update done message to  
digitalairlines  
Jul  4 11:55:52 da12 dhcpd: Update request all from  
digitalairlines: nothing pending  
Jul  4 11:55:52 da12 dhcpd: failover peer digitalairlines:  
peer update completed.  
Jul  4 11:55:52 da12 dhcpd: failover peer digitalairlines:  
I move from recover to recover-done  
Jul  4 11:55:52 da12 dhcpd: failover peer digitalairlines:  
peer moves from recover to recover-done  
Jul  4 11:55:52 da12 dhcpd: failover peer digitalairlines:  
I move from recover-done to normal  
Jul  4 11:55:52 da12 dhcpd: failover peer digitalairlines:  
peer moves from recover-done to normal  
Jul  4 11:55:52 da12 dhcpd: pool 800e40f8 10.0.0/24 total  
20 free 20 backup 0 lts 10  
Jul  4 11:55:52 da12 dhcpd: pool response: 10 leases
```

When a client requests an IP address, you will see messages like this in `/var/log/messages`:

```
Jul  4 11:59:26 da10 dhcpd: DHCPREQUEST for 192.168.5.16
from 00:04:ac:d6:58:96 via eth0: ignored (not
authoritative).
Jul  4 11:59:30 da10 dhcpd: DHCPREQUEST for 192.168.5.16
from 00:04:ac:d6:58:96 via eth0: ignored (not
authoritative).
Jul  4 11:59:36 da10 dhcpd: pool 800e4138 10.0.0/24 total
20 free 10 backup 10 lts 0
Jul  4 11:59:36 da10 dhcpd: DHCPDISCOVER from
00:04:ac:d6:58:96 via eth0
Jul  4 11:59:36 da10 dhcpd: DHCPREQUEST for 10.0.0.111
(10.0.0.12) from 00:04:ac:d6:58:96 via eth0: lease owned by
peer
Jul  4 11:59:37 da10 dhcpd: DHCPPOFFER on 10.0.0.110 to
00:04:ac:d6:58:96 (linux-5ncs) via eth0
```

In this case, the client send a DHCPREQUEST in order to get same the IP address as the last time (192.168.5.16). This address is not available from a range definition, so the server refuses to offer this address. The the client send a DHCPDISCOVER to detect a DHCP server.

The next message from the client is a DHCPREQUEST for the IP address 10.0.0.111, this address is provided from the secondary server (“lease owned by peer”). The last message is the DHCPPOFFER from the secondary server.

On the secondary server, the corresponding messages look like this:

```
Jul  4 12:00:20 dal2 dhcpd: DHCPREQUEST for 149.44.85.16
from 00:04:ac:d6:58:96 via eth1: ignored (not
authoritative).
Jul  4 12:00:25 dal2 dhcpd: DHCPREQUEST for 149.44.85.16
from 00:04:ac:d6:58:96 via eth1: ignored (not
authoritative).
Jul  4 12:00:30 dal2 dhcpd: pool 800e40f8 10.0.0/24 total
20 free 10 backup 10 lts 0
Jul  4 12:00:30 dal2 dhcpd: DHCPDISCOVER from
00:04:ac:d6:58:96 via eth1
Jul  4 12:00:31 dal2 dhcpd: DHCPOFFER on 10.0.0.111 to
00:04:ac:d6:58:96 (linux-5ncs) via eth1
Jul  4 12:00:31 dal2 dhcpd: DHCPREQUEST for 10.0.0.111
(10.0.0.12) from 00:04:ac:d6:58:96 (linux-5ncs) via eth1
Jul  4 12:00:31 dal2 dhcpd: DHCPACK on 10.0.0.111 to
00:04:ac:d6:58:96 (linux-5ncs) via eth1
```

On this server, the DHCPOFFER message is printed as this server offers the IP address. The final message is DHCPACK from the client.

The DHCP log file `/var/lib/dhcp/db/dhcpd.leases` contains information like this:

```
failover peer "digitalairlines" state {
    my state recover at 2 2006/07/04 09:57:39;
    partner state unknown-state at 2 2006/07/04 09:57:39;
}

failover peer "digitalairlines" state {
    my state recover at 2 2006/07/04 09:57:39;
    partner state recover at 2 2006/07/04 09:57:39;
}

failover peer "digitalairlines" state {
    my state recover-done at 2 2006/07/04 09:57:56;
    partner state recover at 2 2006/07/04 09:57:39;
}

failover peer "digitalairlines" state {
    my state recover-done at 2 2006/07/04 09:57:56;
    partner state recover-done at 2 2006/07/04 09:57:39;
}

failover peer "digitalairlines" state {
    my state normal at 2 2006/07/04 09:57:56;
    partner state recover-done at 2 2006/07/04 09:57:39;
}

failover peer "digitalairlines" state {
    my state normal at 2 2006/07/04 09:57:56;
    partner state normal at 2 2006/07/04 09:57:39;
}

lease 10.0.0.120 {
    starts 2 2006/07/04 09:57:56;
    tstp 2 2006/07/04 09:57:56;
    binding state backup;
}

lease 10.0.0.119 {
    starts 2 2006/07/04 09:57:56;
    tstp 2 2006/07/04 09:57:56;
    binding state backup;
}

...
```



```
...
lease 10.0.0.120 {
    starts 2 2006/07/04 09:57:56;
    tstp 2 2006/07/04 09:57:56;
    tsfp 2 2006/07/04 09:57:56;
    binding state backup;
}
lease 10.0.0.119 {
    starts 2 2006/07/04 09:57:56;
    tstp 2 2006/07/04 09:57:56;
    tsfp 2 2006/07/04 09:57:56;
    binding state backup;
}
...
lease 10.0.0.111 {
    starts 2 2006/07/04 10:00:31;
    ends 2 2006/07/04 10:30:31;
    tstp 2 2006/07/04 09:57:56;
    tsfp 2 2006/07/04 10:45:31;
    cltt 2 2006/07/04 10:00:31;
    binding state active;
    next binding state expired;
    hardware ethernet 00:04:ac:d6:58:96;
    uid "\001\000\004\254\326X\226";
}
```

The file `/var/lib/dhcp/db/dhcpd.leases` on the secondary contains information like the following:

```
failover peer "digitalairlines" state {
    my state recover at 2 2006/07/04 09:58:50;
    partner state unknown-state at 2 2006/07/04 09:58:50;
    mclt 0;
}

failover peer "digitalairlines" state {
    my state recover at 2 2006/07/04 09:58:50;
    partner state recover at 2 2006/07/04 09:58:50;
    mclt 1800;
}

failover peer "digitalairlines" state {
    my state recover-done at 2 2006/07/04 09:58:51;
    partner state recover at 2 2006/07/04 09:58:50;
    mclt 1800;
}

failover peer "digitalairlines" state {
    my state recover-done at 2 2006/07/04 09:58:51;
    partner state recover-done at 2 2006/07/04 09:58:50;
    mclt 1800;
}

failover peer "digitalairlines" state {
    my state normal at 2 2006/07/04 09:58:51;
    partner state recover-done at 2 2006/07/04 09:58:50;
    mclt 1800;
}

failover peer "digitalairlines" state {
    my state normal at 2 2006/07/04 09:58:51;
    partner state normal at 2 2006/07/04 09:58:50;
    mclt 1800;
}

lease 10.0.0.120 {
    starts 2 2006/07/04 09:57:56;
    tsfp 2 2006/07/04 09:57:56;
    binding state backup;
}
```

When the secondary server fails (e.g. the server is shut down) and recovers later again, messages like the following will appear in `/var/log/messages` on the primary:

```
Jul  4 12:16:25 da10 dhcpd: peer digitalairlines:
disconnected
Jul  4 12:16:25 da10 dhcpd: failover peer digitalairlines:
I move from normal to communications-interrupted
Jul  4 12:20:06 da10 dhcpd: failover peer digitalairlines:
peer moves from normal to normal
Jul  4 12:20:06 da10 dhcpd: failover peer digitalairlines:
I move from communications-interrupted to normal
Jul  4 12:20:06 da10 dhcpd: pool 800e4138 10.0.0/24 total
20 free 10 backup 9 lts 0
```

On the secondary, the startup messages look like this:

```
Jul  4 12:21:01 da12 dhcpd: failover peer digitalairlines:
I move from normal to startup
Jul  4 12:21:01 da12 dhcpd: failover peer digitalairlines:
peer moves from normal to communications-interrupted
Jul  4 12:21:01 da12 dhcpd: failover peer digitalairlines:
I move from startup to normal
Jul  4 12:21:01 da12 dhcpd: failover peer digitalairlines:
peer moves from communications-interrupted to normal
Jul  4 12:21:01 da12 dhcpd: pool 800e40f8 10.0.0/24 total
20 free 10 backup 9 lts 0
```



In order to not get confused about the time stamps, make sure that you synchronize the time stamps of all your servers using the network time protocol (ntp).

---

The client log file `/var/lib/dhpcd/dhpcd-eth0.info` does only contain the information about the client's configuration and the DHCP server that provided the information:

```
IPADDR=10.0.0.111
NETMASK=255.255.255.0
NETWORK=10.0.0.0
BROADCAST=10.0.0.255
GATEWAY=10.0.0.254
DOMAIN='digitalairlines.com'
DNS=10.0.0.254
DHCPSSID=10.0.0.12
DHCPGIADDR=0.0.0.0
DHCPsiADDR=0.0.0.0
DHCPCHADDR=00:04:AC:D6:58:96
DHCPshADDR=00:04:AC:D6:55:F4
DHCPsNAME=' '
LEASETIME=1800
RENEWALTIME=900
REBINDTIME=1575
INTERFACE='eth0'
CLASSID='Linux 2.6.16.20-0.12-default i686'
CLIENTID=00:04:AC:D6:58:96
```

## Summary

Objective	Summary
1. Configure DHCP Pools	<p>The <b>pool</b> declaration can be used to specify a pool of addresses that will be treated differently than any other pool of addresses, even on the same network segment or subnet.</p>
2. Configure DHCP Failover	<p>The failover protocol allows two DHCP servers to share a common address pool.</p> <p>If one server fails, the other server will continue to renew leases out of the pool and will allocate new addresses.</p> <p>The failover protocol defines a primary server role and a secondary server role.</p> <p>In order to configure failover, you need to write a peer declaration in the file <code>/etc/dhcpd.conf</code> that configures the failover protocol, and you need to write peer references in each pool declaration for which you want to do failover.</p>



## SECTION 11 Manage OpenLDAP

OpenLDAP is the most popular open source LDAP suite. It provides not only the LDAP server itself, but also applications and tools to control and query the server and to develop LDAP-based software.

### Objectives

1. Install and Set Up an OpenLDAP Server
2. Activate LDAP Authentication
3. Replicate OpenLDAP Servers

## Objective 1    Install and Set Up an OpenLDAP Server

To install and set up an OpenLDAP server, you need to do the following:

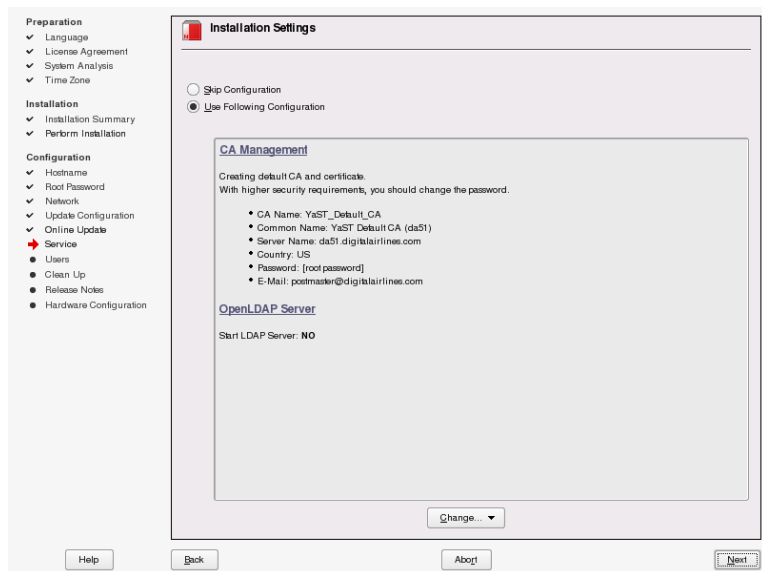
- Install the Required Software and Start the Server
- Configure OpenLDAP with YaST
- Edit the OpenLDAP Configuration Files

### ***Install the Required Software and Start the Server***

Normally, you can set up an OpenLDAP server with YaST during the installation process of SUSE Linux Enterprise Server 10.

Select **OpenLDAP Server** in the Installation Settings dialog.

**Figure 11-1**





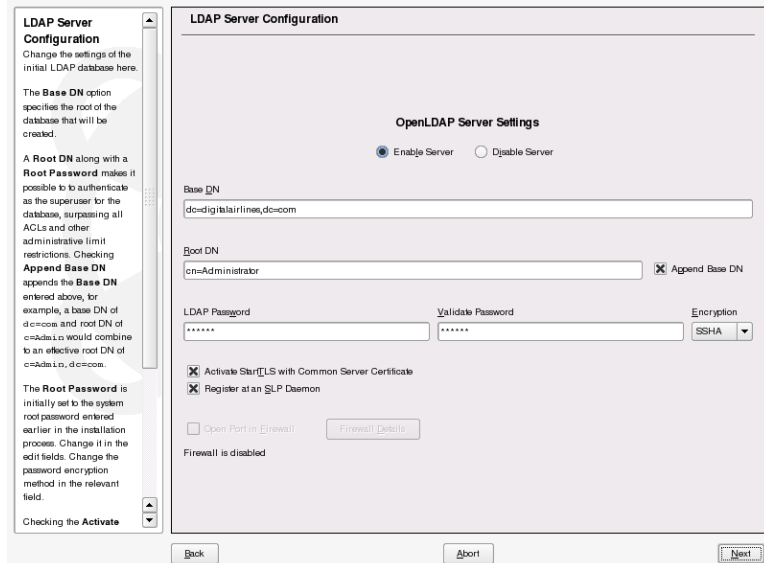
The important values are entered automatically in the text fields. A warning informs you that any change will disable the automatic generation of base DN, root DN, and the LDAP password.

**Figure 11-2**



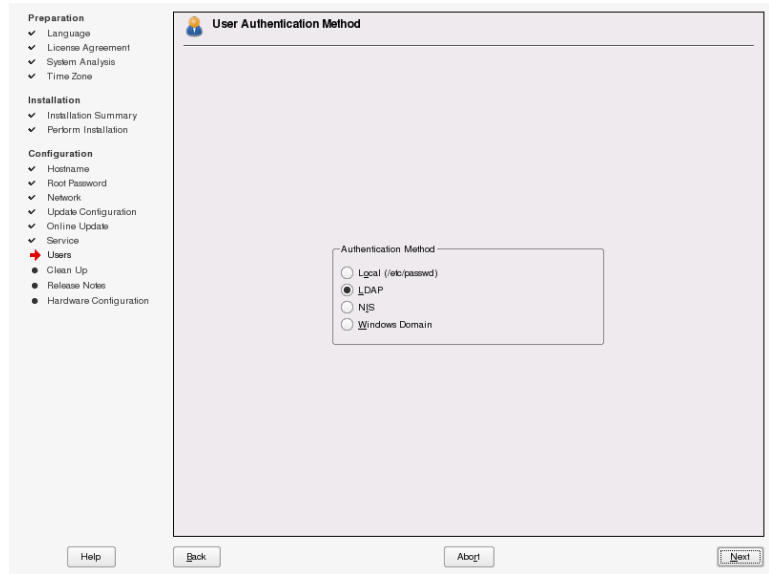
Select **OK** and enable the OpenLDAP server by selecting **Enable Server**.

**Figure 11-3**



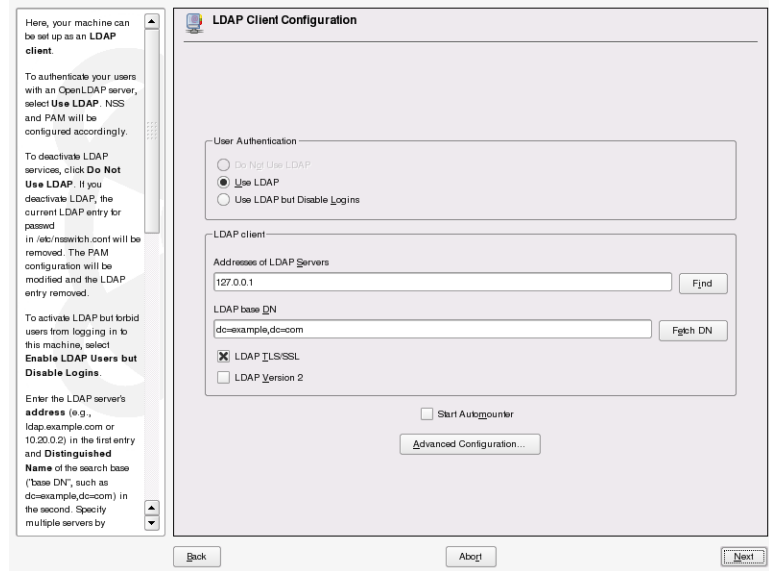
If you want to authenticate via LDAP, select **LDAP** in the **User Authentication Method** dialog.

**Figure 11-4**



In the next dialog, you can enter the configuration of your LDAP client.

**Figure 11-5**



Here you can enter the following information:

- **User Authentication.** Select the kind of user authentication:
  - ❑ **Do Not Use LDAP.** LDAP is not used for user authentication.
  - ❑ **Use LDAP.** LDAP is used for user authentication.
  - ❑ **Use LDAP but Disable Logins.** LDAP is used for user authentication, but the users are not able to log in to this machine anymore.
- **Addresses of LDAP Servers.** Enter the name or the IP address of your LDAP server here. If the IP of your LDAP server is provided via SLP, you can select **Find**.

- **LDAP base DN.** Enter the DN of the root of your LDAP tree here. Select **Fetch DN** if you want to browse the LDAP tree of your LDAP server.
- **LDAP TLS/SSL.** If your LDAP server supports encryption, activate this option.
- **LDAP Version 2.** The current LDAP protocol version is 3. If you want to use the older version 2, select this option.
- **Start Automounter.** Starts the automounter daemon which mounts directories automatically when they are used.

### ***Configure OpenLDAP with YaST***

However, if you chose not to install the server during installation, you can set up an LDAP server by installing the following software packages with YaST:

- `openldap2`
- `openldap2-client`
- `pam_ldap`
- `nss_ldap`

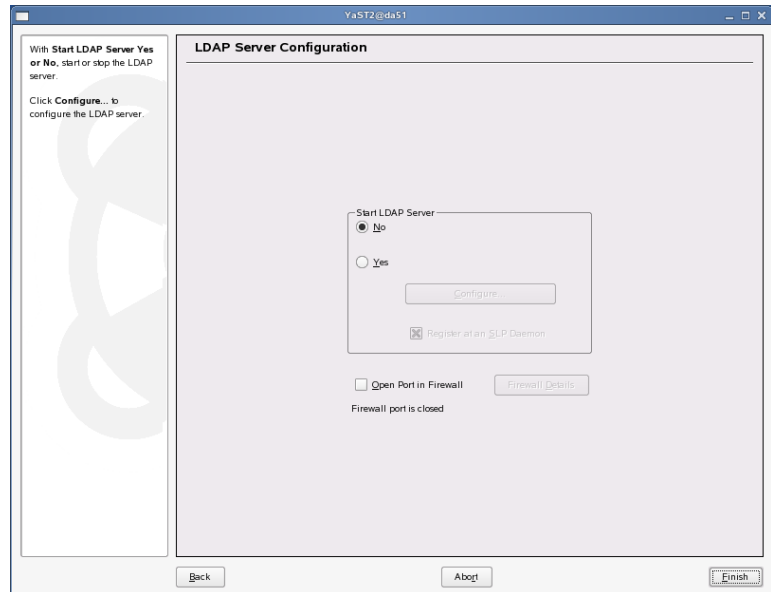
In YaST two modules concerning OpenLDAP are available:

- YaST Module for the LDAP Server
- YaST Module for the LDAP Client

## YaST Module for the LDAP Server

If you select **LDAP Server** in the YaST **Network Services** section, but your LDAP server is currently not running, the following dialog appears.

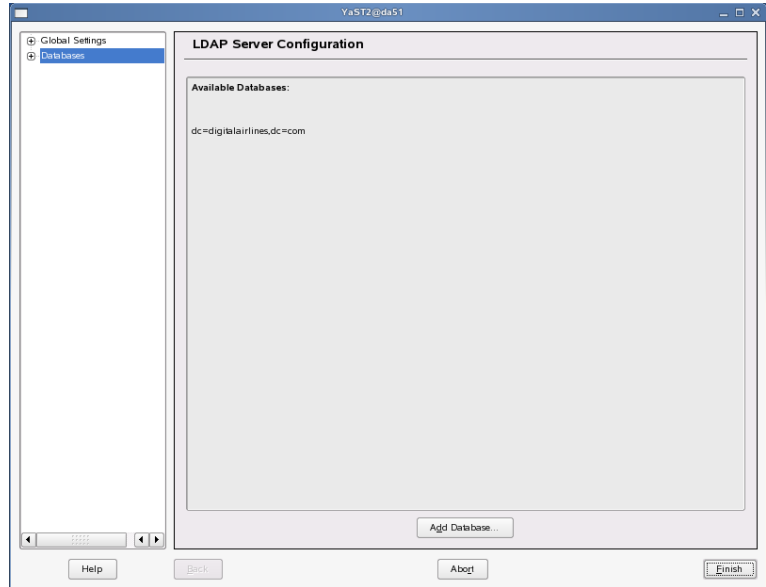
**Figure 11-6**



Select **Yes** to start and configure the LDAP server. You can also activate the option to register the LDAP server at a SLP daemon or to open the LDAP port in the firewall.

Select **Configure** to configure the LDAP server.

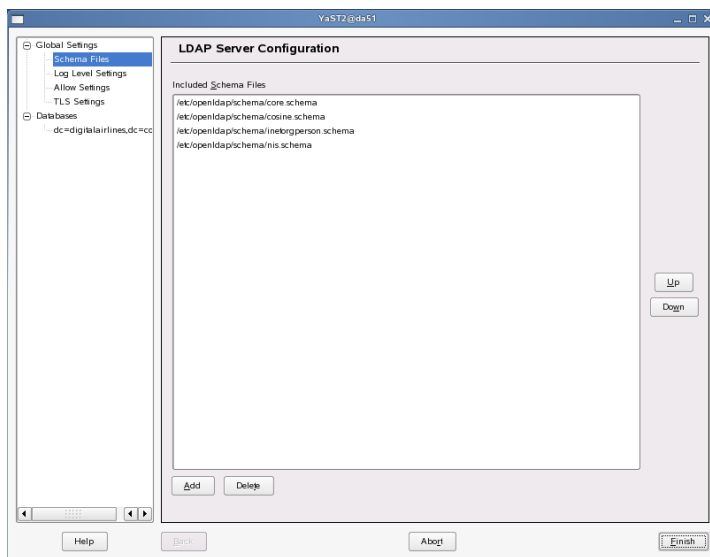
**Figure 11-7**



Open the **Global Settings** submenu in the left frame. The following settings can be configured:

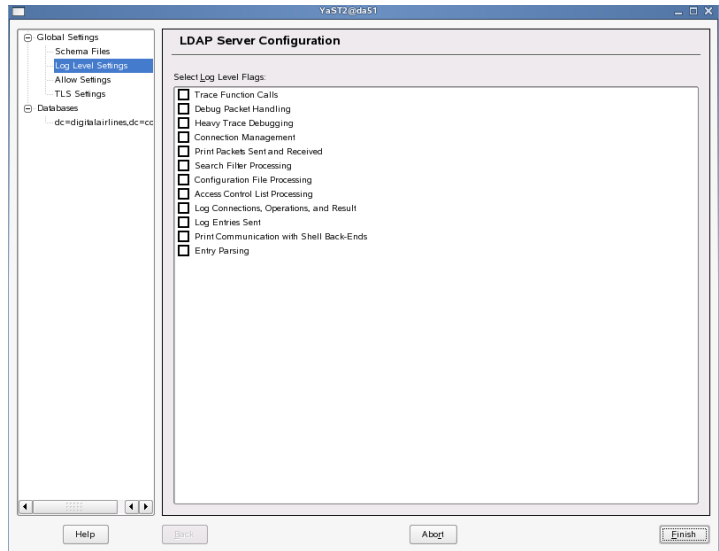
- **Schema Files.** Add or remove LDAP schema files using **Add** or **Delete**.

**Figure 11-8**



- **Log Level Settings.** Select the information you want to log.

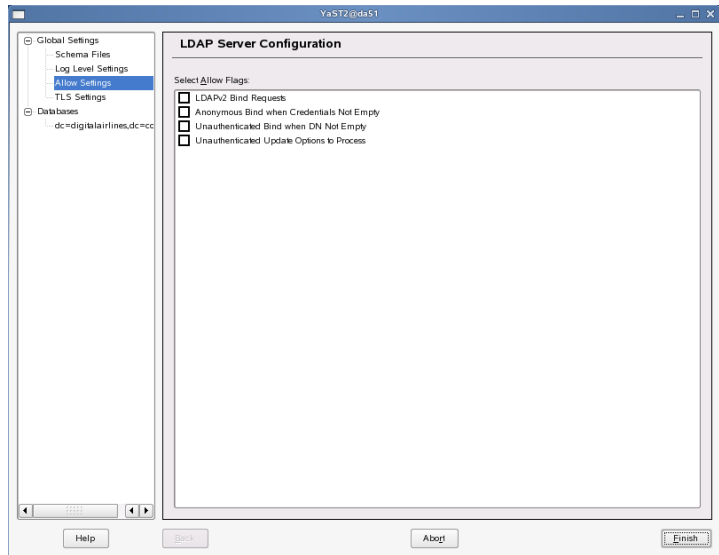
**Figure 11-9**





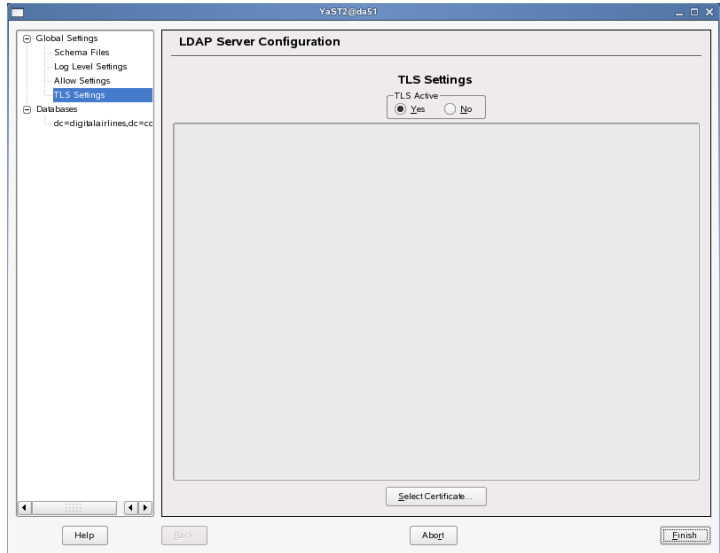
- **Allow Settings.** Select options concerning access to stored information.

**Figure 11-10**



- **TLS Settings.** Activate or Deactivate TLS.

**Figure 11-11**



If you select **Select Certificate**, you can import a certificate.



To create a new certificate use the YaST module **Security and Users > CA Management**.

If you select **Databases** in the left frame, you can add a new database by selecting Add Database. The following dialog appears.

**Figure 11-12**

In this dialog you have to enter the following information:

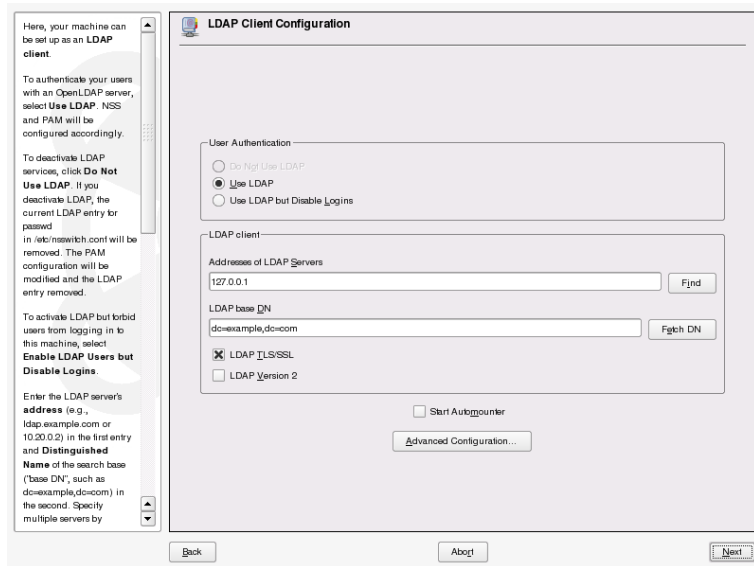
- **Base DN.** DN of the root of the LDAP tree.
- **Root DN.** DN of the administrator. If you activate **Append Base DN**, the content of **Base DN** is appended to the input of **Root DN** automatically.
- **LDAP Password, Validate Password.** The administrator's password. From the **Encryption** menu you can select the encryption method that should be used to encrypt the password.
- **Database Directory.** Path to the directory the database is stored in.

Most of the information of the YaST LDAP server are stored in the file `/etc/sysconfig/openldap`.

## YaST Module for the LDAP Client

When selecting **LDAP Client** in the YaST Network Services section, the following dialog appears as described above.

**Figure 11-13**



This information is stored in the file `/etc/sysconfig/ldap`. Three variables are used:

- **BASE\_CONFIG\_DN**. DN of the root of the LDAP tree.
- **BIND\_DN**. The DN of the administrator.
- **FILE\_SERVER**.

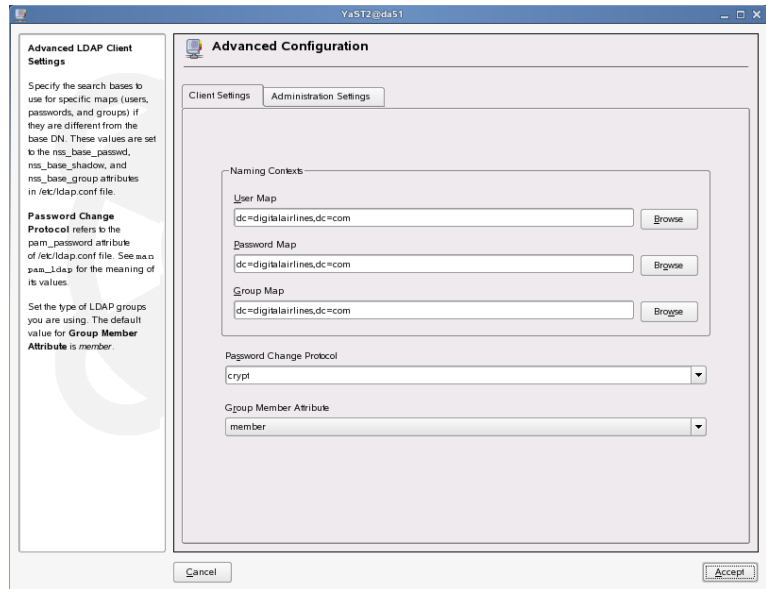
Selecting the **Advanced Configuration** button a dialog with two tabs appears:

- Client Settings
- Administrator Settings

## Client Settings

In this dialog you can specify the search base for users, passwords, and groups.

**Figure 11-14**



The items of the dialog are:

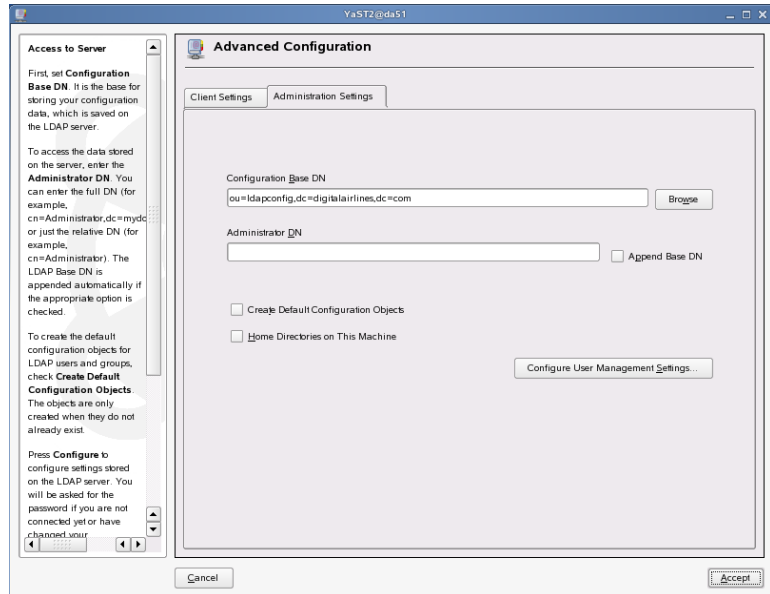
- **User Map.** Enter the DN where users are stored.
- **Password Map.** Enter the DN where passwords are stored.
- **Group Map.** Enter the DN where groups are stored.
- **Password Change Protocol.** The password change protocol. See **man 5 pam\_ldap** to get more information about the possible values.
- **Group Member Attribute.** Set the type of the LDAP groups you are using. The value can depend on the schema files you are using. The default is **member**.

This information is stored in the file `/etc/ldap.conf`.

## Administrator Settings

In this dialog, you specify some administrator settings.

**Figure 11-15**



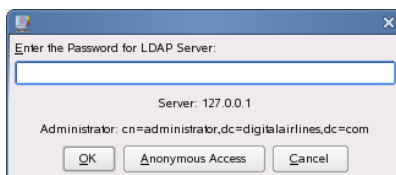
Enter the following information:

- **Configuration Base DN.** The configuration data are also stored in the LDAP directory. Here you can enter the base DN for all configuration information.
- **Administrator DN.** The DN of the LDAP administrator.
- **Create Default Configuration Objects.** To create the default configuration objects for users and groups, select this option.

- **Home Directories on This Machine.** If home directories of users should be stored on your machine, select this option. This option is only information for the YaST user module that manages user home directories.

If you want to configure the user management settings stored on the LDAP server (e.g., the next free UID), select **Configure User Management Settings**. You are asked to enter the administrator's LDAP password.

**Figure 11-16**



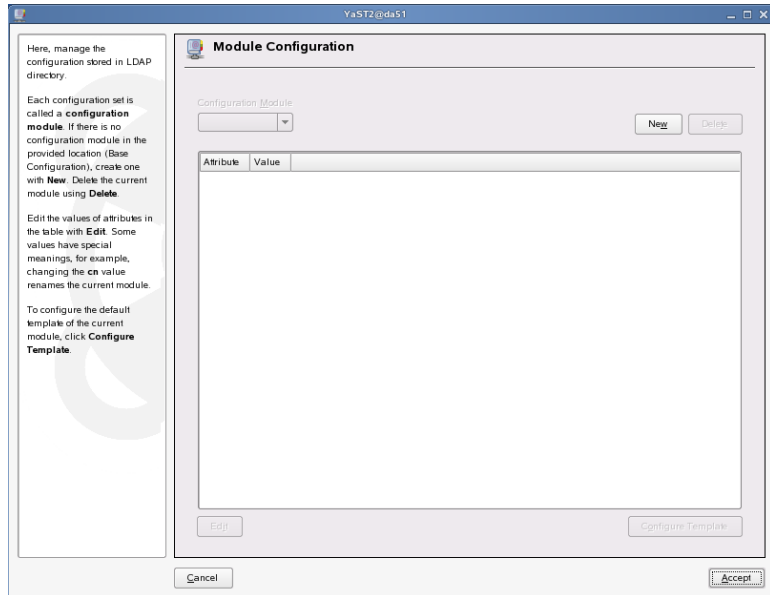
After you enter the administrator password, YaST determines if the configuration base DN exists in the LDAP directory. If not, it asks if the entry should be created.

**Figure 11-17**



Select **Yes** and the Module Configuration dialog appears.

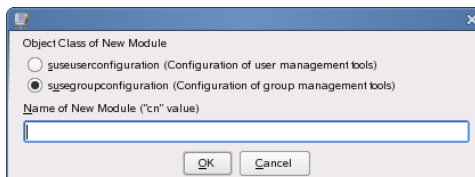
**Figure 11-18**



A configuration module is a set of configurations. You can manage configuration modules for user and group configurations.

Select **New** to create a new module.

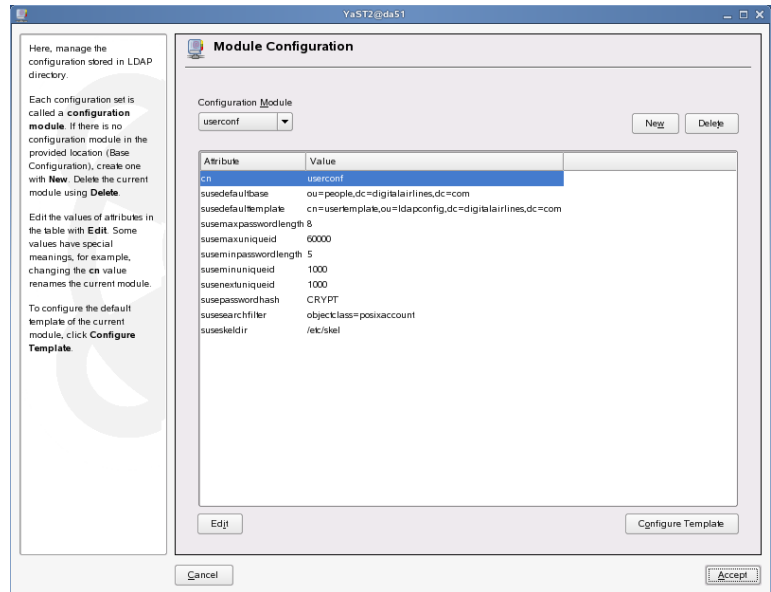
**Figure 11-19**





Select whether you want to create a module for user configuration or for group configuration. You also have to enter the common name (cn) of the module.

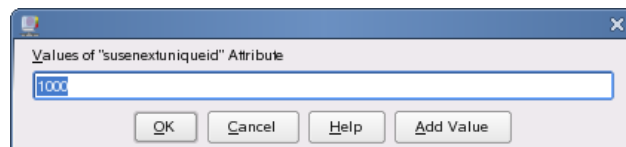
**Figure 11-20**



A template for the user or group module is shown. You can find the listed attributes in the YaST schema file (/etc/openldap/schema/yast.schema).

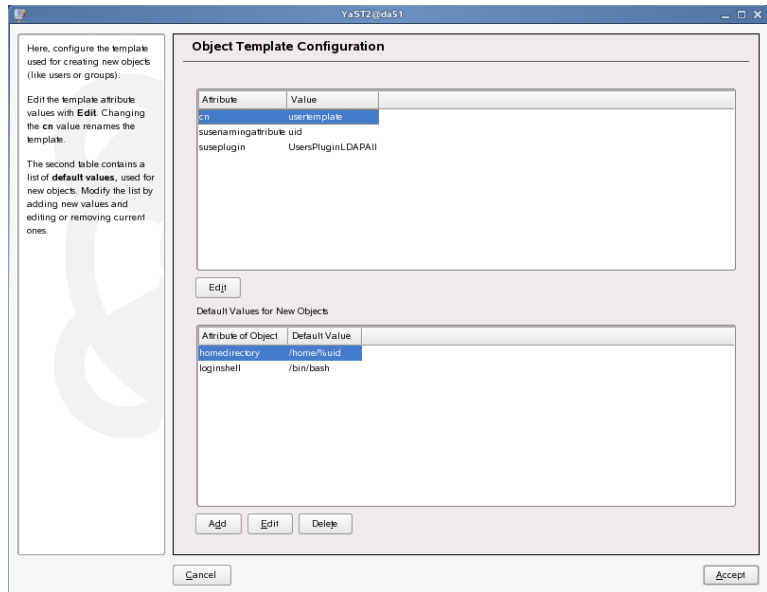
To edit a value of an attribute, select a line from the list and then select **Edit**.

**Figure 11-21**



In the line of the attribute **susedefaulttemplate**, a template for new users or groups is specified. If you want to edit this template, select **Configure Template**.

**Figure 11-22**



In the upper frame you see a list of attributes and their values. Edit the template attribute values with **Edit**.

In the lower frame you can specify default values for attributes. Use the three buttons below as follows:

- **Add.** Add a new default value.
- **Edit.** Edit the selected default value.
- **Delete.** Delete the selected default value.

## **Exercise 11-1     Set Up OpenLDAP with YaST**

In this exercise, you set up an OpenLDAP server and client using YaST. The base DN is “dc=digitalairlines,dc=com” and the common name of the LDAP administrator is “cn=Administrator,dc=digitalairlines,dc=com” with password “novell”. TLS/SSL is not used in this exercise.

Do the following:

- Part I - Install OpenLDAP
- Part II - Setup the OpenLDAP Server
- Part III - Setup the OpenLDAP Client

### **Part I - Install OpenLDAP**

1. From the main menu, start **YaST**.
2. Enter the root password (**novell**) and select **OK**.
3. From the YaST Control Center, select **Software > Software Management**.
4. From the filter drop-down menu, select **Search**.
5. In the Search field, enter **ldap**; then select **Search**.
6. On the right, select the following packages:
  - **nss\_ldap**
  - **openldap2**
  - **openldap2-client**
  - **pam\_ldap**
7. Select **Accept**; then insert the requested *SUSE Linux Enterprise Server 10* DVD.
8. When installation is complete, remove the DVD and close the YaST Control Center.

## Part II - Setup the OpenLDAP Server

1. Start YaST.
2. Start the YaST module **Network Services > LDAP Server**.
3. Select **Yes** to start the LDAP server.
4. Select **Configure**.
5. In the LDAP Server Configuration dialog select **Add Database** to add a database.
6. In the Add Database dialog enter the following information:

Table 11-1

Textbox	Value
Base DN	<b>dc=digitalairlines,dc=com</b>
LDAP Password	<b>novell</b>
Validate Password	<b>novell</b>

7. Select **OK**.
8. In the left frame select **Global Settings > TLS Settings**. Make Sure that the option TLS Active is set to **no**.
9. Select **Finish**.

## Part III - Setup the OpenLDAP Client

1. Start the YaST module **Network Services > LDAP Client**.
2. Select **Use LDAP** to activate LDAP for user authentication.
3. Make sure that the content of Addresses of LDAP Servers is **127.0.0.1**.
4. Make sure that the content of LDAP base DN is **dc=digitalairlines,dc=com**
5. Make sure that the option **LDAP TLS/SSL** is deactivated.

6. Select **Advanced Configuration**.
7. Select the **Administration Settings** tab.
8. Enter **cn=Administrator** in the textbox Administration DN.
9. Activate the option **Append Base DN**.
10. Activate the option **Create Default Configuration Objects**.
11. Activate the option **Home Directories on This Machine**.
12. Select **Accept**.
13. Select **Finish**.
14. When asked to enter the administrator password enter **novell**.

***(End of Exercise)***

## ***Edit the OpenLDAP Configuration Files***

The configuration files for OpenLDAP are located in the directory `/etc/openldap/`. The directory contains two configuration files:

- Configure the OpenLDAP Server with `slapd.conf`
- Configure the LDAP Clients with `ldap.conf`

### **Configure the OpenLDAP Server with `slapd.conf`**

The default version of the configuration file `/etc/openldap/slapd.conf` has only a few options. Most of them do not need to be changed:

- **include**. The first entry includes a schema file. In this schema file some object classes and their attributes are defined. You can use these object classes to build a database.

For simple example configurations, only the object classes of the *core* schema needs to be included.

- **referral**. Specifies the referral to pass back when the slapd cannot find a local database to handle a request.
- **pidfile**. Path to the file that stores the process ID.
- **argsfile**. Path to the file that stores the server's command line options.
- **modulepath**. Backend modules that are loaded dynamically. With the **modulepath** you specify a list of directories to search for loadable modules.
- **moduleload**. Specifies the filename or the absolute path of a dynamically loadable module.
- **security**. Specifies a set of security strength factors to require. The directive may be specified globally and/or per-database. Possible security strength factors are:
  - **ssf**. Specifies the overall security strength factor.

- ❑ **transport**. Specifies the transport security strength factor.
- ❑ **tls**. Specifies the TLS security strength factor.
- ❑ **sasl**. Specifies the SASL security strength factor.
- ❑ **update\_\***. Specifies the security strength factors required for directory updates.
- ❑ **simple\_bind**. Specifies the security strength factor required for simple username/password authentication.

For a more detailed description, read the section about `sasl-secprops` in the `slapd.conf` manpage (**man 5 slapd.conf**).

- **access to**. Grants access to a set of entries and/or attributes by one or more requesters.

Syntax: **access to *objects* [by *requesters level control*]**

The *objects* and the *requesters* are specified by their DN's.

As *requesters*, you also can use the asterisk for *everybody*. **self** means the owner object of the attribute. You can also specify a DN by **dn=DN** or a group by **group=DN**.

The *level* can be:

- ❑ **read** (Read access)
- ❑ **write** (Write access)
- ❑ **auth** (Authentication needed)
- ❑ **none** (No access)

Example 1:

```
access to attrs=userpassword
    by self write
    by anonymous auth
    by * none

access to *
    by * read
```

## Example 2:

```
access to *  
  by dn="cn=Administrator,dc=digitalairlines,dc=com" write  
  by * read
```

- **database.** Specifies the kind of back end. The following databases are supported:
  - ❑ **bdb.** Berkeley DB
  - ❑ **dnssrv.** DNS SRV
  - ❑ **hdb.** A hierarchical variant of bdb
  - ❑ **ldap.** Lightweight Directory Access Protocol (Proxy)
  - ❑ **ldbm.** Lightweight DBM
  - ❑ **meta.** Meta directory
  - ❑ **monitor.** Monitor backend
  - ❑ **passwd.** Read-only access to passwd
  - ❑ **perl.** Perl programmable backend
  - ❑ **shell.** External shell program
  - ❑ **sql.** SQL programmable backend

- **suffix “DN”**

Specify the DN suffix of queries that will be passed to this backend database.

Example:

```
suffix "dc=digitalairlines,dc=com"
```

- **checkpoint *ops* *minutes*.** Checkpoints are only tested after successful write operations. If *ops* operations have been completed or more than the specified number of *minutes* have passed, a new checkpoint is performed.



- **cachesize**. Size of in-memory cache of the LDBM backend database.

- **rootdn “DN”**

This line sets the administrator of the LDAP server. You can also configure the domain components in this line.

Example:

```
rootdn "ou=slc,dc=digitalairlines,dc=com"
```

- **rootpw *password***

This line specifies the password for the administrator. The default password secret must be changed.

For security reasons, the password should be stored in an encrypted form. To create an encrypted password, use the following command:

**slappasswd -s *your\_password***

The command outputs a string that has to be copied into the configuration file.

```
da51:~ # slappasswd -s novell  
{SSHA}5JHiJ3Q17MpxaNT95DqVg2u1VfjZyzHh
```

The entry for the command rootpw looks like the following:

```
rootpw "{SSHA}5JHiJ3Q17MpxaNT95DqVg2u1VfjZyzHh"
```

- **directory**. Path to the BDB or HDB database. This directory must exist before the server is started.
- **index**. Specifies the indices to maintain. This increases the speed of the search.

Syntax: **index *attributes types***

Possible *types* are:

- ❑ **pres.** Allows filters to be used that ask if the attribute is present in an entry (cn=\*).
- ❑ **eq.** Allows filters to be used that ask if an attribute has an exact value. It includes presence, so it is not necessary to index something as pres,eq.
- ❑ **approx.** Allows filters to be used that ask if an attribute value is “similar to” something.
- ❑ **sub.** Allows filters to be used that do substring searches on an attribute's values.
- ❑ **none.**

After finishing the configuration, you can start the server with the following command:

### **rcldap start**

If you want to start the LDAP server automatically when the server boots, use the following command:

### **insserv ldap**

## **Configure the LDAP Clients with ldap.conf**

After you have changed the server configuration file, you need to change the client configuration. Two files are important:

- /etc/openldap/ldap.conf
- /etc/ldap.conf

### **/etc/openldap/ldap.conf**

The file `/etc/openldap/ldap.conf` is the configuration file of the LDAP client.

The most important options are specified below:

- **host 127.0.0.1**

This line sets the default server that LDAP clients should connect to.

- **base dc=digitalairlines,dc=com**

This is the default directory search base that should be used by LDAP clients.

After a standard installation, there is one more line in `/etc/ldap.conf`:

```
TLS_REQCERT      allow
```

`TLS_REQCERT` specifies what checks, if any, to perform on server certificates in a TLS session. The level can be specified as one of the following keywords:

- **never**. The client will not request or check any server certificate.
- **allow**. The server certificate is requested. If no certificate or a bad certificate is provided, the session proceeds normally.
- **try**. The server certificate is requested. If no certificate is provided, the session proceeds normally. If a bad certificate is provided, the session is immediately terminated.
- **demand** or **hard**. The server certificate is requested. If no certificate or a bad certificate is provided, the session is immediately terminated. (default)

For more options, read the manual **man 5 ldap.conf**.

### **/etc/ldap.conf**

The file `/etc/ldap.conf` is the configuration file for the LDAP nameservice switch library, the LDAP PAM module, and the shadow package. The file belongs to the RPM package `pwdutils`.

```
da51:~ # rpm -qf /etc/ldap.conf
pwdutils-3.0.7.1-17.10
da51:~ #
```

The most important options of `/etc/ldap.conf` are:

- **host**. Specifies the name(s) or IP address(es) of the LDAP server(s) to connect to.
- **base**. Specifies the default base distinguished name (DN) to use for searches.
- **ldap\_version**. Specifies the version of the LDAP protocol; “3” stands for LDAPv3.
- **bind\_policy**. Specifies what happens if the LDAP server is not reachable. The possible values are:
  - **hard**. `nss_ldap` will retry connecting to the software with exponential backoff. (default)
  - **soft**. Forbids `nss_ldap` from retrying failed LDAP queries.
- **binddn**. Specifies the distinguished name with which to bind to the directory server(s).

The default is to bind anonymously.

- **bindpw**. Specifies the cleartext credentials with which to bind. This option is only applicable when used with `binddn` above.
- **scope**. Specifies the search scope. Possible values are:
  - **sub**. subtree
  - **one**. one level
  - **base**. base object

The default scope is **sub**; **base** scope is almost never useful for nameservice lookups.

- **pam\_password**. Specifies the password change protocol to use. The following protocols are supported:
  - **clear**. Change password using an LDAPModify request, replacing the userPassword value with the new cleartext password.
  - **clear\_remove\_old**, **nds** or **racf**. Change password using an LDAPModify request, first removing the userPassword value containing the old cleartext password, and then adding the userPassword value with the new cleartext password. This protocol is necessary for use with Novell NDS and IBM RACF.
  - **crypt**. Change password using an LDAPModify request, first generating a one-way hash of the new password using crypt(3) and then replacing userPassword value with the new hashed password.
  - **md5**. Change password using an LDAPModify request, first generating a one way hash of the new password using MD5 and then replacing userPassword value with the new hashed password.
  - **ad**. Change password using an LDAPModify request, using the Active Directory Services Interface (ADSI) password change protocol.
  - **exop**. Change password using the RFC 3062 password modify extended operation (only the new password is sent).
  - **exop\_send\_old**. Change password using the RFC 3062 password modify extended operation (both the old and new passwords are sent). This is the preferred choice when using the PADL XAD identity server.
- **ssl**. Specifies whether to use SSL/TLS or not. The following values are possible:
  - **on**

- ❑ **off** (default)
- ❑ **start\_tls**

If **start\_tls** is specified, then StartTLS is used rather than raw LDAP over SSL.



---

Not all LDAP client libraries support both SSL and StartTLS, and all related configuration options.

---

- **nss\_map\_attribute** *from\_attribute to\_attribute*. This option may be specified multiple times, and directs `nss_ldap` to use the attribute *to\_attribute* instead of the RFC 2307 attribute *from\_attribute* in all lookups.

For example: **nss\_map\_attribute**    **uniqueMember member**

- **pam\_login\_attribute**. Specifies the attribute to use when constructing the attribute value assertion for retrieving a directory entry for a user's login name.

The default is “uid”, for compatibility with RFC 2307.

- **pam\_filter**. Specifies a filter to use when retrieving user information. The user entry must match the **pam\_login\_attribute** attribute value as well as any filter specified here.

For example: **pam\_filter**    **objectclass=posixAccount**

- **nss\_base\_map**. Specify the search base, scope, and filter to be used for specific maps.



---

Note that **map** forms part of the configuration file keyword and is one of the following options **passwd**, **shadow**, **group**, **hosts**, **services**, **networks**, **protocols**, **rpc**, **ethers**, **netmasks**, **bootparams**, **aliases**, and **netgroup**.

---

The syntax of **basedn** and **scope** are the same as for the configuration file options of the same name, with the addition of being able to omit the trailing suffix of the base DN (in which case the global base DN will be appended instead).



---

For more options, read the manual **man 5 pam\_ldap**.

---

## Objective 2      **Activate LDAP Authentication**

To activate authentication via LDAP you have to do the following:

- Change the User Password
- Activate pam\_ldap

### ***Change the User Password***

If you use the LDAP directory for user authentication, you can change the user's password by using the **ldappasswd** command.

```
da51:~ # ldappasswd -x -D
"cn=Administrator,dc=digitalairlines,dc=com" -W -S
"uid=geeko,ou=people,dc=digitalairlines,dc=com"
New password:
Re-enter new password:
Enter LDAP password:
Result: Success (0)
da51:~ #
```

First you have to enter the new password for the user twice. Then you have to enter the administrator's password to fulfill the change.



The user can change his or her own password by:

```
da51:~ # ldappasswd -x -D
"uid=geeko,ou=people,dc=digitalairlines,dc=com" -W -S
"uid=geeko,ou=people,dc=digitalairlines,dc=com"
New password:
Re-enter new password:
Enter LDAP password:
Result: Success (0)
da51:~ #
```



---

If you want to create a home directory automatically when the user logs in the first time, add the following line to `/etc/pam.d/sshd`

**session required pam\_mkhomedir.so skel=/etc/skel/ umask=0022**

---

If the user now searches for the new user's LDAP entry he should be prompted to enter his password and the crypted password should be shown in the search results.

```
da51:~ # ldapsearch -x -D
"uid=geeko,ou=people,dc=digitalairlines,dc=com" -W
Enter LDAP Password:
# extended LDIF
#
# LDAPv3
# base <> with scope subtree
# filter: uid=geeko
# requesting: ALL
#
# geeko, people, digitalairlines.com
dn: uid=geeko,ou=people,dc=digitalairlines,dc=com
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
uid: geeko
uidNumber: 1000
gidNumber: 100
cn: Geeko Chameleon
givenName: Geeko
sn: Chameleon
homeDirectory: /home/geeko
loginShell: /bin/bash
shadowMax: 99999
shadowWarning: 7
shadowInactive: -1
shadowMin: 0
shadowLastChange: 12609
userPassword::
e1NTSEF9TVJjWDlHQm8ydGxIUE1HWWFUT2lOQWZlNDkrUHA4OTU=

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
da51:~ #
```

## ***Activate pam\_ldap***

To activate PAM and NSS, you have to add LDAP as a possible source for authentication. You have to make sure the following lines are in file `/etc/nsswitch.conf`:

```
passwd: compat
group:  compat
...
passwd_compat:  ldap
group_compat:   ldap
```

If you configure the LDAP client with YaST the PAM module `pam_ldap.so` is activated for all kinds of authentication. This is done in the file `/etc/security/pam_unix2.conf`.

```
auth:      use_ldap
account:    use_ldap
password:   use_ldap
session:    none
```

If you configure the LDAP client manually, you can activate the LDAP authentication for selected services.

To do this, use the PAM module `pam_ldap.so` (included in the package `pam_ldap`) is used. Add the module corresponding to the service to the PAM configuration file in `/etc/pam.d`.

For example, if you want LDAP authentication for `ssh`, the file `/etc/pam.d/sshd` must look like this:

```
##PAM-1.0
auth      include      common-auth
auth      required      pam_nologin.so
account   include      common-account
password  include      common-password
session   include      common-session
# Enable the following line to get resmgr support for
# ssh sessions (see /usr/share/doc/packages/resmgr/README)
#session  optional      pam_resmgr.so fake_ttyname
```

Here the files `common-auth`, `common-account`, `common-password` and `common-session` are included. These files can also be found in `/etc/pam.d/`. The file `/etc/pam.d/common-auth` has -for example- the following statements:

```
auth    required    pam_env.so
auth    required    pam_unix2.so
```

If a required module fails, the modules following the failed module are tested anyway. But an error occurs at the end. If a sufficient module fails, but another sufficient module matches, login is allowed.

There are a couple of examples available in the directory `/usr/share/doc/packages/pam_ldap/pam.d/`.

The advantage of using LDAP for all kinds of authentication (as done by YaST) is that you can use standard tools like **passwd** or **chsh**.

```
kbailey@da51:~> passwd
Changing password for kbailey.
Enter login(LDAP) password:
New password:
Re-enter new password:
LDAP password information changed for geeko
```

## **Exercise 11-2     Set up an LDAP User Database**

In this exercise, you add a user with the following information to your LDAP directory:

**Table 11-2**

Information	Value
Login	kbailey
Name	Kate Bailey
UID	1010
GID	100
Home directory	/home/kbailey
Login shell	/bin/bash

You create a password (“novell”) for user kbailey. In part II you enable an automatically creation of the home directory if a user logs in using **ssh**. In part III you login as user kbailey to test your configurations.

Do the following:

- Part I - Add Users to the LDAP Directory
- Part II - Set LDAP User Password
- Part III - Configure Automatically Home Directory Creation
- Part IV - Login as kbailey via SSH

### **Part I - Add Users to the LDAP Directory**

1. There should be some entries in the LDAP directory created by YaST. To see the content of your LDAP directory enter **ldapsearch -x**
2. Create an LDIF file by entering **vi example.ldif**

3. To create a Posix account for a new user kbailey, add the following lines:

```
dn: uid=kbailey,ou=people,dc=digitalairlines,dc=com  
objectClass: posixAccount  
objectClass: shadowAccount  
objectClass: inetOrgPerson  
uid: kbailey  
uidNumber: 1010  
gidNumber: 100  
cn: Kate Bailey  
givenName: Kate  
sn: Bailey  
homeDirectory: /home/kbailey  
loginShell: /bin/bash  
shadowMax: 99999  
shadowWarning: 7  
shadowInactive: -1  
shadowMin: 0  
shadowLastChange: 12609
```

You also can copy the file `exercises/section_3/kbailey.ldif` from the Course DVD.

4. Save the file and exit the text editor by entering **:wq**.
5. Add the LDIF file to your LDAP database by entering  
**ldapadd -x -D**  
**"cn=Administrator,dc=digitalairlines,dc=com" -W -f**  
**kbailey.ldif**
6. Enter the password for the LDAP administrator (**novell**).
7. To see the content of your LDAP directory, enter  
**ldapsearch -x "uid=kbailey"**

## Part II - Set LDAP User Password

1. Use the command **getent** (Get Entry) to test whether the LDAP database is used by the glibc. Enter

**getent passwd**

The users in `/etc/passwd` and the LDAP database should be listed. The end of the output may look similar to this:

```
...
uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash
wwwrun:x:30:8:WWW daemon apache:/var/lib/wwwrun:/bin/false
geeko:x:1000:100:Geeko Chameleon:/home/geeko:/bin/bash
kbailey:*:1010:100:Kate Bailey:/home/kbailey:/bin/bash
```

2. To set the password for the user `kbailey`, enter  
**ldappasswd -x -D**  
**"cn=Administrator,dc=digitalairlines,dc=com" -W -S**  
**"uid=kbailey,ou=people,dc=digitalairlines,dc=com"**
3. Enter **novell** twice as the new password.
4. Enter the administrator password (also **novell**) to set the password.
5. Search for your own LDAP entry by entering  
**ldapsearch -x -D**  
**"cn=Administrator,dc=digitalairlines,dc=com" -x**  
**"uid=kbailey" -W**
6. Enter the administrator password (**novell**). Your encrypted user password should be listed.

### Part III - Configure Automatically Home Directory Creation

1. Open the file `/etc/pam.d/sshd` with `vi` by entering  
**`vi /etc/pam.d/sshd`**
2. Add the following line  
**`session required pam_mkhomedir.so skel=/etc/skel/  
umask=0022`**
3. Save the file and exit by entering **`:wq`**.

### Part IV - Login as kbailey via SSH

1. Login as user kbailey by entering  
**`ssh -l kbailey localhost`**
2. Enter **yes** to modify the list of known hosts.
3. Enter the password of kbailey (**novell**).  
The home directory should be created.
4. Enter **exit** to log out.

***(End of Exercise)***



## Objective 3      Replicate OpenLDAP Servers

If there are a lot of requests for the LDAP server you may need more than one server. One server is the master. The other servers are slaves. A master/slave arrangement provides an effective way to increase capacity, availability, and reliability.

Changes on the master server are propagated by the slurpd to the slaves. In this objective we cover the following topics:

- Add the Replicaton DN to the LDAP Directory
- Configure slapd for Replication
- The Command-Line Options of slurpd
- Transfer the LDAP Database

### ***Add the Replicaton DN to the LDAP Directory***

In the LDAP directory (on master and slave) there must be an object with the DN used for replication. This can look like this:

```
dn: uid=replicator,dc=digitalairlines,dc=com
objectClass: inetOrgPerson
uid: replicator
cn: LDAP Replicator
sn: Replicator
```

You also have to set a password for this object using **ldappasswd**.

## Configure slapd for Replication

The slapd on the master server can be configured to write a replication logfile by the following entry /etc/openldap/slapd.conf:

```
repllogfile /var/lib/ldap/master-slapd.repllog
```

This file contains regular LDIF change records, e.g.

```
time: 1148659780
dn: ou=groups,dc=digitalairlines,dc=com
changetype: add
objectClass: organizationalUnit
ou: groups
structuralObjectClass: organizationalUnit
entryUUID: c7b0c4e0-811d-102a-9c4d-2129ebe1b381
creatorsName: cn=Administrator,dc=digitalairlines,dc=com
createTimestamp: 20060526160940Z
entryCSN: 20060526160940Z#000000#00#000000
modifiersName: cn=Administrator,dc=digitalairlines,dc=com
modifyTimestamp: 20060526160940Z
```

On the master server you have to specify the name(s) of the slave server(s) with the **replica** option:

```
replica uri=ldap://10.0.0.51:389
binddn="cn=replicator,dc=digitalairlines,dc=com" bindmethod=simple
credentials=novell
```

The following parameters are used:

- **uri.** Name or IP of the slave server. The default LDAP port is 389. If you want to use LDAPS (port 636), the uri looks like this  
**replica uri=ldaps://10.0.0.51:636 ...**
- **binddn.** DN of the master server that is allowed to access the slave. This DN must also be specified on the slave server with the option **updatedn.**



---

This DN should not be the same as the master's rootdn.

---

- **bindmethod.** Specifies the authentication method. You can select between **simple** (password-based authentication) and **sasl** (SASL authentication).
- **credentials.** Password for simple authentication.

On the slave server you have to add the **updatedn** option, as mentioned. For example:

```
updatedn="cn=replicator,dc=digitalairlines,dc=com"
```

Use the **updateref** directive to define the URL the slave should return if an update request is received.

```
updateref="ldap://10.0.0.51"
```

You have to make sure that this DN has permission to write the database (e.g., by **access** directives).

```
access to *  
  by dn="cn=replicator,dc=digitalairlines,dc=com" write  
  by * read
```

## ***The Command-Line Options of slurpd***

The man page of slurpd can be accessed by **man 8 slurpd**.

The options of **slurpd** are:

- **-d level**. Specify the debug level. Possible debug levels are:
  - **4**. Heavy trace debugging
  - **64**. Configuration file processing
  - **65535**. Enable all debugging
  - **?**. Lists all debug levels and exits slurpd.
- **-f file**. Specifies the slapd configuration file. The default is `/etc/openldap/slapd.conf`.
- **-r file**. Specifies the name of the slapd replication logfile.
- **-o**. Run in "one-shot" mode. Normally, slurpd processes the `repllog` file and then watches for more replication entries to be appended. In one-shot mode, slurpd processes a replication log and exits.
- **-t directory**. slurpd copies the replication log to a working directory before processing it.

There is a start script available in `/etc/init.d/` and you can start the slurpd by entering **rcslurpd start**.

## ***Transfer the LDAP Database***

Before you start the slapd on the slave server, you have to transfer the LDAP database from the master to the server.

Make sure that the slapd of the master is shut down. To transfer the database (/var/lib/ldap/ by default), you can use **scp**.

Alternatively you can use the command **slapcat**. **slapcat** is used to generate an LDIF output based upon the contents of a slapd database. By default the output is written to STDOUT. You can specify a file by using the option **-l**.

slapcat has other interesting options as well (e.g., for filtering). See the man page for further details (**man 8 slapcat**).

## **Exercise 11-3      Replicate OpenLDAP Servers**

In this exercise, you configure LDAP replication with your neighbor.

You use a new LDAP entry  
“uid=replicator,dc=digitalairlines,dc=com” with password “novell”  
for replication.

In part IV, you change the surname of user kbailey from “Bailey” to  
“Smith”. You do this on the master to see if the replication works  
correct.

Do the following:

- Part I - Add the Replicator DN to the Master’s LDAP Directory
- Part II - Configure the LDAP Master Server
- Part III - Configure the LDAP Client Server
- Part IV - Copy the Database and Start the Servers
- Part V - Test Replication

### **Part I - Add the Replicator DN to the Master’s LDAP Directory**

1. To add the new LDAP entry on the master create a new LDIF file  
by entering  
**vi replicator.ldif**
2. Enter the following lines  
**dn: uid=replicator,dc=digitalairlines,dc=com**  
**objectClass: inetOrgPerson**  
**uid: replicator**  
**cn: LDAP Replicator**  
**sn: Replicator**

You also can copy the file exercises/section\_3/replicator.ldif  
from the Course DVD.

3. Save the file and exit by entering **:wq**.
4. Add the entry to the LDAP directory by entering  
**ldapadd -x -D**  
**"cn=Administrator,dc=digitalairlines,dc=com" -W -f replicator.ldif**
5. To enter a password for the replicator enter  
**ldappasswd -x -D**  
**"cn=Administrator,dc=digitalairlines,dc=com" -W -S**  
**"uid=replicator,dc=digitalairlines,dc=com"**
6. Enter **novell** twice as new password.
7. Enter the administrator password (also **novell**) to set the password.

## Part II - Configure the LDAP Master Server

1. On the master stop your LDAP server by entering **rcldap stop**.
2. Open the file `/etc/openldap/slapd.conf` by entering  
**vi /etc/openldap/slapd.conf**
3. Specify a file where the changes in the LDAP directory are stored by adding the line  
**replogfile /var/lib/ldap/master-slapd.repllog**
4. Specify the replication host (all in one line)  
**replica uri=ldap://slave\_ip:389**  
**binddn="uid=replicator,dc=digitalairlines,dc=com"**  
**bindmethod=simple credentials=novell**  
*slave\_ip* is the IP address of your neighbor's computer.
5. Save the file and exit by entering **:wq**.

### Part III - Configure the LDAP Client Server

1. Stop your LDAP server by entering **rldap stop**.
2. Open the file `/etc/openldap/slapd.conf` by entering **vi /etc/openldap/slapd.conf**
3. Add the following lines  
**updatedn="uid=replicator,dc=digitalairlines,dc=com"**  
**updateref="ldap://master\_ip"**  
*master\_ip* is the IP address of your neighbor's computer.
4. Change the following access rule  
**access to \***  
**by \* read**  
  
to  
**access to \***  
**by dn="uid=replicator,dc=digitalairlines,dc=com" write**  
**by \* read**
5. Save the file and exit by entering **:wq**.

### Part IV - Copy the Database and Start the Servers

1. On the master server enter  
**scp -r /var/lib/ldap/ host:/var/lib/ldap/**  
Replace *host* by the host name of the slave server.
2. Enter **novell** when prompted for a password.
3. On both servers start the slapd by entering  
**rldap start**
4. On the master server start the slurpd by entering  
**rcslurpd start**



## Part V - Test Replication

1. To change some user information on the master, create a new LDIF file by entering

**vi change.ldif**

2. Enter the following lines

**dn: uid=kbailey,ou=people,dc=digitalairlines,dc=com**

**cn: Kate Smith**

**sn: Smith**

You also can copy the file exercises/section\_3/change.ldif from the Course DVD.

3. Save the file and exit by entering **:wq**.

4. To change the LDAP information, enter

**ldapmodify -x -D**

**“cn=Administrator,dc=digitalairlines,dc=com” -W -f  
change.ldif**

5. On the master and slave enter

**ldapsearch -x “uid=kbailey”**

***(End of Exercise)***

## Summary

Objective	Summary
1. Install and Set Up an OpenLDAP Server	<p>If you did not configure an OpenLDAP server during the installation, you need to install the following software packages:</p> <ul style="list-style-type: none"><li>■ openldap2</li><li>■ openldap2-client</li><li>■ pam_ldap</li><li>■ nss_ldap</li></ul> <p>The configuration of the OpenLDAP server is located in the file <code>/etc/openldap/slapd.conf</code>.</p> <p>You can create passwords for the administrator entry of the configuration file with the command <b>slappasswd</b>.</p> <p><code>/etc/openldap/ldap.conf</code> is the configuration file of the LDAP client.</p> <p><code>/etc/ldap.conf</code> is the configuration file for the LDAP nameservice switch library, the LDAP PAM module, and the shadow package.</p>

Objective	Summary
2. Activate LDAP Authentication	<p>If you use the LDAP directory for user authentication, you can change the user's password by using the <b>ldappasswd</b> command.</p> <p>If you configure the LDAP client with the YaST module, the PAM module <code>pam_ldap.so</code> is activated for all kinds of authentication in the file <code>/etc/security/pam_unix2.conf</code>.</p> <p>If you configure the LDAP client manually, you can activate the LDAP authentication only for selected services.</p> <p>To do this, use the PAM module <code>pam_ldap.so</code> (included in the package <code>pam_ldap</code>) is used. Add the module corresponding to the service to the PAM configuration file in <code>/etc/pam.d</code>.</p>

---

Objective	Summary
3. Replicate OpenLDAP Servers	<p>The slapd on the master server can be configured to write a replication logfile.</p> <p>On the master server you also have to specify the name(s) of the slave server.</p> <p>On the master you specify a DN to used by the master to access the slave. This DN must also be specified on the slave server and must have permission to write to the database.</p> <p><b>slapcat</b> is used to generate an LDIF output based on the contents of a slapd database.</p>

## SECTION 12 Configure a Mail Server

The standard mail server in SUSE Linux Enterprise Server 10 is Postfix.

In this section you will learn some backgrounds about the SMTP Protocol. Tools to detect spam mails and viruses are also introduced.

### Objectives

1. Understand SMTP Communication
2. Manage Spam
3. Use a Virus Scanner for Email

## Objective 1      Understand SMTP Communication

In the following we want to have a closer look at:

- The SMTP Commands
- Command Syntax
- SMTP Reply Codes
- Minimal SMTP Command Implementation
- An Example for Sending Mail with Telnet

### *The SMTP Commands*

The commands are:

- **HELO** (Hello) is used by the SMTP sender to open a connection to an SMTP recipient (a greeting) and to introduce itself with its full host name. This host name is passed as an argument attached to the **HELO** command.  
  
The SMTP recipient answers this greeting with its full host name. This completes the initialization of communication between the SMTP sender and the recipient.
- **MAIL FROM** initializes transmission of an email message. In the simplest case, the **MAIL FROM** command takes the sender's email address (reverse-path) as an argument. The argument can be optionally extended to include a list of hosts in front of the recipient's address via which a reply should be routed (source route relaying), e.g.:
  - @venus.example.com
  - @mars.example.com:Sales@example.com
- **RCPT TO** (Recipient) sets the recipient's address (forward-path) for an item of mail. If the mail should be sent to several recipients at the same time, this command is repeated that number of times.

The command may also include a list of hosts (e.g., @mars.example.com, @venus.example.com:info@example.com) via which the mail should be routed (source route relaying).

If the argument contains source route relaying through a list of hosts, the mail will be routed directly to the first host given. This host removes itself from the beginning of the list and forwards the mail to the next host in the list.

- **DATA** tells the SMTP mailer that anything that follows is the content of the mail. The end of the mail content is indicated by Enter.

When the SMTP recipient recognizes the end of the mail transmission, it begins to process the information received. This involves interpreting and possibly updating the forward and reverse paths. If this data processing results in no failures, the SMTP sender is sent an **OK**. Otherwise, an error message is sent.

A time stamp is inserted at the beginning of the mail as soon as the SMTP recipient receives a mail to forward to a further SMTP recipient or for direct delivery to the recipient. This time stamp contains information about the identity of the SMTP sender, the SMTP recipient, and the time when the mail reached the SMTP recipient.

Mails that are routed over several SMTP relays contain a corresponding number of these time stamps. If the SMTP recipient is responsible for the ultimate delivery of the mail (if, for example, the addressee's mailbox is located on the SMTP recipient), the current contents of the return path (i.e., the contents of the **MAIL FROM** argument after being modified through the previous SMTP relays) are inserted at the top of the mail.

- **VRFY** (Verify) verifies a user ID. This causes the SMTP recipient to check the validity of the addressee given as an argument. If the SMTP reader knows the addressee, the address is expanded into a full address (including domain). The **VRFY** command has no influence on the **MAIL FROM**, **RCPT TO**, and **DATA** instructions.
- **EXPN** (Expand) instructs the SMTP recipient to treat the argument as a mailing list. As a result, the members of the list are transmitted to the SMTP sender.
- **RSET** (Reset ) resets all the information previously stored about the SMTP recipient. The forward path, reverse path, and mail contents are lost.
- **HELP** takes as its argument any other SMTP command. A page of tips about how to use the corresponding instruction will be displayed.
- **NOOP** (No Operation) causes the SMTP recipient to answer with an **OK**. Apart from that, no changes are made to the mail content or the forward and return paths.
- **QUIT** ends the connection between the SMTP sender and recipient.



Command Syntax

SMTP commands provide direct communication between the sender and recipient. The most commonly used SMTP commands are described in the following table.

Table 12-1

Command Syntax	Description of the Arguments
HELO <i>hostname</i>	<i>hostname</i> contains the complete host name of the SMTP sender.
MAIL FROM: <i>reverse-path</i>	<i>reverse-path</i> contains the relay path for source route relaying and the sender's address.
RCPT TO: <i>forward-path</i>	<i>forward-path</i> contains the relay path for source route relaying and the recipient's address.
DATA <i>data</i>	<i>data</i> constitutes the contents of the mail (the actual message).
RSET	./.
VERFY <i>string</i>	<i>string</i> contains a user or mailbox name.
EXPN <i>string</i>	<i>string</i> contains a mailing list identifier.
HELP [ <i>string</i> ]	<i>string</i> contains any SMTP command.
NOOP	./.
QUIT	./.

Commands can be written in lower case or upper case; SMTP commands are not case sensitive. For example, the SMTP recipient will treat the following commands in the same way: **MAIL FROM**, **Mail From**, **mail from**, **Mail FrOm**.

In contrast, the arguments in *forward-path* and *reverse-path* may be case-sensitive. The interpretation of these arguments depends on the SMTP recipient's operating system and the structure of the user database, which is why lowercase and uppercase characters may be treated differently.

### **SMTP Reply Codes**

During communication between the SMTP sender and SMTP recipient, the sender transmits various commands to the recipient and controls the course of the communication as a whole. The recipient acknowledges the message's receipt and gives the status of command processing with a corresponding reply code. Only when the sender has received a receipt for the previous command can it start transmitting the next command.

The most commonly found SMTP reply codes are listed in the table below:

**Table 12-2**

<b>Reply Code</b>	<b>Description</b>
211	System status or system help reply.
214	Displays the help message after entering <b>HELP</b> .
220	SMTP server ready.
221	SMTP server has closed connection.
250	Specified command has been carried out.
251	User not local; will forward to <i>forward-path</i> .
354	Start of the DATA entry.
421	SMTP service not available.
450	Requested action not taken (mailbox is already in use).

**Table 12-2** (continued)

Reply Code	Description
451	Requested action aborted.
452	Requested action not taken; insufficient storage space in system.
500	Syntax error, command unrecognized.
501	Syntax error in parameters or arguments.
502	Command not implemented.
503	Bad sequence of commands.
504	Command not implemented for that parameter.
550	Requested action not taken; file unavailable (e.g., mailbox not found, no access).
551	User does not exist on mail server; try <b><i>forward-path</i></b> .
552	Requested mail action aborted: storage allocation exceeded.
553	Requested action not taken: mailbox name not allowed (e.g., mailbox syntax incorrect).
554	Transaction failed.

### ***Minimal SMTP Command Implementation***

Not all commands introduced up to now are implemented by every server. Certain SMTP commands are intentionally not included in some server implementations to increase the security of the server. One such security-critical command is **HELP**. The Postfix SMTP server by Wietse Venema has not fully implemented this command and simply responds to a **HELP** command with reply code 502 ("Command not implemented").

The following list shows the SMTP commands that an SMTP server absolutely must implement to provide SMTP communication:

- **HELO**
- **MAIL FROM:**
- **RCPT TO:**
- **DATA**
- **RSET**
- **NOOP**
- **QUIT**

## ***An Example for Sending Mail with Telnet***

The text below shows a typical example of the log output that can easily be reconstructed using a telnet session on port 25 (standard mail server port):

```
da53:~ # telnet da51.digitalairlines.com 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 da51.digitalairlines.com ESMTP Postfix
HELO da53.digitalairlines.com
250 da51.digitalairlines.com
MAIL FROM: jgoldman@digitalairlines.com
250 Ok
RCPT TO: geeko@digitalairlines.com
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Subject: Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed diem nonummy nibh euismod tincidunt ut laoreet dolore
magna aliquam erat volutpat. Ut wisis enim ad minim veniam,
quis nostrud exerci tution ullamcorper suscipit lobortis
isl ut aliquip ex ea commodo consequat.
.
250 Ok: queued as 45B5116A9D
QUIT
221 Bye
Connection closed by foreign host.
da53:~ #
```

## Objective 2    Manage Spam

This objective explains the following:

- Use SpamAssassin
- Test SpamAssassin

### *Use SpamAssassin*

**spamassassin** (and **spamc**) expect their input from STDIN.

The simplest way to use SpamAssassin is to pipe the mailbox into the command **spamassassin**:

```
da51:~ # cat /var/spool/mail/root | spamassassin
```

If you want to use SpamAssassin with your Postfix configuration, the easiest way is to use Procmail. You need to edit only three files:

- Create the file `/etc/procmailrc` with this content:

```
#LOGFILE=/tmp/procmail.log
#VERBOSE=yes

# Until now, mail is untagged, you may add rules for
# mail that must not be tagged

:0 hbfw
| /usr/bin/spamc
```

The mail is piped in the first filter rule to the **spamc**. The flags of the filter rule are explained below:

- **h.** Feed the header to the pipe, file, or mail destination (default).
- **b.** Feed the body to the pipe, file, or mail destination (default).'

- ❑ **f.** Consider the pipe as a filter.
- ❑ **w.** Wait for the filter or program to finish and check its exitcode (normally ignored); if the filter is unsuccessful, then the text will not have been filtered.
- Make sure that you activate Procmail in `/etc/postfix/main.cf`:

```
mailbox_command = /usr/bin/procmail
```

- Make sure that your smtpd definition in the file `/etc/postfix/master.cf` is set to default

```
smtp      inet  n       -       n       -       -       smtpd
```

After this, start spamd by entering

### **rcspamd start**

Spam can be filtered from the mail client or in a further Procmail configuration (`~/.procmailrc`) by adding lines similar to this:

```
:0
* ^X-Spam-Status: Yes
$MAILDIR/Spam
```

### ***Test SpamAssassin***

You can use **telnet** to test your configuration. You also can send a spam email directly using the **sendmail** command:

```
cat sample-spam.txt | /usr/sbin/sendmail geeko@digitalairlins.com
```

## Objective 3    Use a Virus Scanner for Email

Communication via email is very important for companies and individuals today, but email can be infected by virus software.

Most of the viruses attack Windows clients, but the mail server of a company is often a Linux or UNIX machine. To avoid damage we recommend to search for viruses before the infected mail arrives at the user's client.

SUSE Linux Enterprise Server 10 provides tools to detect viruses in email on your mail server.

In this objective, the following tools are introduced:

- AVMailGate
- AMaViSd-new

### ***AVMailGate***

AVMailGate is the abbreviation for AntiVir MailGate, an antivirus mail filter from H+BEDV Datentechnik GmbH (<http://www.hbedv.com>).

AVMailGate updates the virus definition file and the engine itself.

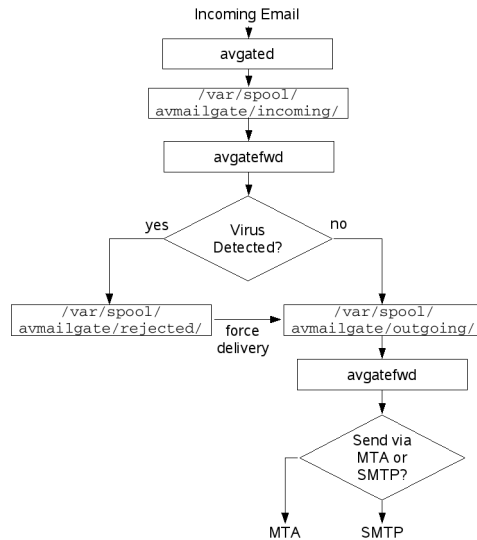
This topic describes the following:

- AVMailGate Architecture
- Install AVMailGate
- Configure AVMailGate
- Understand the AVMailGate Configuration Files
- Update AVMailGate



## AVMailGate Architecture

Figure 12-1



AVMailGate is composed of three queues and two kinds of processes:

- **Queues** (located in /var/spool/avmailgate/)
  - ❑ **incoming/**. The input queue for all incoming email.
  - ❑ **rejected/**. The queue where possible infected email are stored.
  - ❑ **outgoing/**. The output queue for not infected email.
- **Processes**
  - ❑ **avgated**. The smtpd receiver that stores incoming email in the input queue. (daemon)
  - ❑ **avgatefwd**. Virus-scanning function and SMTP forwarder and sendmail invoker. (daemon)

Both processes can be configured by editing the file **/etc/avmailgate.conf**.

This file includes comments that describe the possible settings.



---

For a description of all options, see  
`/usr/share/packages/avmailgate/doc/MANUAL`.

---

## Install AVMailGate

To install AVMailGate, select the package **avmailgate**.

A startscript is available in **rcavgate** to start and stop AVMailGate.

To start AVMailGate at the system's start, enter **insserv avgate**.

## Configure AVMailGate

There are two possibilities to use AVMailGate with Postfix:

- AVMailGate Waits for Mails on Port 25 and Forwards Them to Postfix
- AVMailGate Is Used from Postfix as Content Filter

### AVMailGate Waits for Mails on Port 25 and Forwards Them to Postfix

To use AVMailGate as mail proxy that waits for mails on port 25 you simply have to avoid that Postfix listens on this port. Therefore mark the following line in `/etc/postfix/master.cf`: as command by adding a hash (“#”) in front

```
#smtp      inet  n       -       n       -       -       smtpd
```

Then restart Postfix.

### AVMailGate Is Used from Postfix as Content Filter

The best way to use AVMailGate with Postfix is to use the Full Content Filter API (see `FILTER_README` in the Postfix package for details).

It waits for SMTP connections on port 10024 and sends scanned email messages or virus warnings out per SMTP on port 10025.

To integrate AVMailGate as a filter in Postfix, do the following:

1. Edit `/etc/postfix/master.cf` and uncomment –if applicable– or add the following line:

```
localhost:10025 inet n - y - - smtpd -o  
content_filter=
```

2. Edit `/etc/postfix/main.cf` and add the following line:

```
content_filter = smtp:127.0.0.1:10024
```

3. Since AVMailGate listens on port 10024, not port 25, edit `/etc/avmailgate.conf` and set

```
ListenAddress    localhost    port 10024
```

4. You have to tell AVMailGate it should send email back to Postfix via SMTP on host localhost via port 10025.

Edit `/etc/avmailgate.conf` and set

```
ForwardTo        SMTP: localhost port 10025
```

5. Since AVMailGate sends out notification messages as AVMailGate, set an alias in `/etc/aliases`:

```
vmailgate:      root
```

You must run **newaliases** afterward.

6. After these changes, enter

**rcpostfix reload**

and

**rcavgate start**

Your system is now ready to scan email.

## Understand the AVMailGate Configuration Files

As mentioned before, the configuration file of AVMailGate is `/etc/avmailgate.conf`. The options of this file can be grouped:

- General Parameters
- Scanning of Files in an Archive
- Handling Envelope Recipient Addresses
- Adding a Notification in the Body of Transmitted Mails
- Other Configuration Files

### General Parameters

- **User** and **Group**. `avgated` and `avgatefwd` run with the permissions of this user and group.
- **Postmaster**. Errors and alert messages are sent to this user.
- **MyHostName**. Hostname of the computer. If not set, it is retrieved by `gethostname`.
- **SpoolDir**. The directory where the queues are stored.
- **AntiVirDir**. Directory where the virus signatures are stored.
- **TemporaryDir**. Temporary directory where email messages will be extracted and scanned.

- **PidDir**. The location of the PID files.
- **LogFile**. Location of the log file.
- **SyslogFacility**. Facility argument for the syslog daemon (default: **mail**).
- **MaxIncomingConnections**. Maximum number of simultaneous connections.
- **SMTPTimeout**. Number of seconds until an SMTP timeout occurs.

More detailed timeouts can be configured by using the following options:

- **SMTPGreetingTimeout**
- **SMTPHeloTimeout**
- **SMTPMailFromTimeout**
- **SMTPRcptTimeout**
- **SMTPDataTimeout**
- **SMTPDataBlockTimeout**
- **SMTPDataPeriodTimeout**
- **MaxMessageSize**. Maximum size of a message in bytes.
- **MaxRecipientsPerMessage**. Maximum number of recipients.
- **MinFreeBlocks**. Number of free file system blocks. If the limit is reached, no more incoming email is accepted. (0=disable feature)
- **MaxForwarders**. Maximum number of forward processes of avgatfwd. The number depends on the quality of the network connection (low network quality > higher value).
- **BlockSuspiciousMime**. If set to **YES**, suspicious MIME email will be blocked.
- **BlockExtensions**. Filename extensions that should be blocked (separated by a semicolon).

- **ExposeRecipientsAlerts, ExposeSenderAlerts, ExposePostmasterAlerts.** Specifies if alerts will be sent to the recipient, sender, or postmaster. The possible values are
  - **NO.** No alerts will be sent.
  - **LOCAL** (not available for **ExposePostmasterAlerts**). Alerts will be sent if the recipient/sender is a local user.
  - **YES.** Alerts will be sent.
- **AlertsUser.** Name of sender of alerts. (Syntax: *username* or *username@domain*)
- **ListenAddress.** Interface and port the SMTP daemon listens on. The default interface 0.0.0.0 means all interfaces.  
  
Syntax: **ListenAddress** *interface* **port** *port*
- **ForwardTo.** Type of mail forwarding.
  - By piping: **ForwardTo** *path\_to\_sendmail*
  - By SMTP: **ForwardTo SMTP:** *host* **port** *port*  
Default: **ForwardTo** /usr/sbin/sendmail -oem -oi
- **RefuseEmptyMailFrom.** If set to **YES**, mails containing a blank sender address will be blocked.
- **PollPeriod.** Interval (in seconds) of queue scanning done by avgatefwd.
- **QueueLifetime.** Maximum time a message can stay in the queue before it will be bounced. (0 = disable feature)
- **ForwarderRetryDelay.** Maximum time between retrying to send a queued message.
- **ThrottleMessageCount.** Number of messages that will be reprocessed in a given time. (Only needed for large queues.)  
  
After reprocessing, the avgatefwd will sleep for **ThrottleDelay** seconds.
- **BounceMessageUser.** Name or mail address of the sender of error messages (e.g., if an email could not be delivered).

### Scanning of Files in an Archive

- **ArchiveMaxRecursion.** Maximum of recursion depth of unpacking and scanning archives. (0 = unlimited depth)
- **ArchiveMaxSize.** Maximum file size (in bytes) of an archive that will be scanned. (0 = unlimited size)
- **ArchiveMaxRatio.** Maximum compression ratio of an archive that will be scanned.
- **BlockSuspiciousArchive.** If set to **YES**, email that reach the limits ArchiveMaxRecursion, ArchiveMaxSize or ArchiveMaxRatio are blocked.
- **BlockEncrypedArchive.** If set to **YES**, email with encryped archives are blocked.
- **BlockUnsupportedArchive.** If set to **YES**, email with archives that cannot be scanned are blocked.
- **BlockOnError.** If set to **YES**, mail that cannot be scanned due to scan timeout or process error are blocked.

### Handling Envelope Recipient Addresses

Source routing is a technique whereby the sender of a packet can specify the route that a packet should take through the network.

The following options concern the source routing used by avgated. For a detailed description, read the AVMailGate documentation.

- **AllowSourceRouting**
- **InEnvelopeAddressesBangIs**
- **InEnvelopeAddressesPercentIs**
- **AcceptLooseDomainName**

### Adding a Notification in the Body of Transmitted Mails

- **AddStatusInBody.** If set to **YES**, a default status text is inserted in the email's body.

If you want to insert your own text, you can specify a file name here. For no status text, enter **NO** (default).

- **MaxMessageSizeStatus.** Specify a message size up to where the status text is added to the message.

Syntax: **MaxMessageSizeStatus Xm | k | b**

- **ForwardAllEmailAsMIME.** If set to **YES**, all incoming non-MIME mails are converted to MIME.
- **AddPrecedenceHeader.** If set to **YES**, each notice mail is maked with "Precedence:" in the header. You also can enter a custom text.
- **AddressFilter.** If set to **YES**, each sender and/or recipient address will be matched against the tables `/etc/avmailgate.scan` and `/etc/avmailgate.ignore`.

The order in which the tables are scanned can be specified in **FilterTableOrder**. (first hit matches)

- **UseProxy.** You can optimize the scans by using the proxy feature in an AVMailGate pool.

The number of anti-virus scanners in the pool can be specified with the **ProxyScanners** option.

**ProxyConnections** specifies the number of simultaneous allowed connections.

- **AddHeaderToNotice.** If set to **YES**, a mail header is added to postmaster notice mails.
- **BounceMessageSizeBody, BounceMessageSizeHeader.** Limit the size (in bytes) of bounced mails.
- **AddXHeaderInfo.** If set to **YES**, the information about the scanning status is added to the header of the checked mail.



- **AddReceivedByHeader.** If set to **YES**, a “Received:” is added to the mail’s header.
- **MaxHopCount.** If there are more than **MaxHopCount** "Received:" lines in the header, the mail will not be accepted.  
Prevent mail loops.

### Other Configuration Files

There are four more configuration files for AVMailGate. These files contain a couple of regular expressions. We will only give a short overview here:

- **/etc/avmailgate.acl.** Defines which hosts are considered local and for which the server is allowed to relay emails.
- **/etc/avmailgate.ignore.** Defines mail addresses that should not be scanned.
- **/etc/avmailgate.scan.** Defines mail addresses that are always scanned.
- **/etc/avmailgate.warn.** In this file one can specify who receives a mail in case of an alert.

### Update AVMailGate

The file that includes the virus signatures is  
`/usr/lib/AntiVir/antivir[0123].vdf`.

To update these files, enter

**`/usr/lib/AntiVir/antivir --update`**

The output looks like the following:

```
AntiVir / Linux Version 2.1.5-24 +gui
Copyright (c) 1994-2005 by H+BEDV Datentechnik GmbH.
All rights reserved.

Warning: the file "antivir.vdf" is more than 14 days old
checking for updates

06.32.00.60 = 06.32.00.60 [vdf database (part 0), on-disk]
06.32.18.16 < 06.34.00.105 [vdf database (part 1), on-disk]
06.32.18.17 < 06.34.00.106 [vdf database (part 2), on-disk]
06.33.00.07 < 06.34.00.124 [vdf database (part 3), on-disk]
06.33.00.11 < 06.34.00.14 [scan engine, running]
06.33.00.11 < 06.34.00.14 [scan engine, on-disk]
antivir1.vdf 100% |*****| 1630 KB 1.59 MB/s 0:00
ETA
antivir2.vdf 100% |*****| 1 KB 0.00 KB/s --:--
ETA
antivir3.vdf 100% |*****| 35 KB 0.00 KB/s --:--
ETA
antivir 100% |*****| 695 KB 0.00 KB/s --:--
ETA
06.32.00.60 = 06.32.00.60 [vdf database (part 0), on-disk]
06.34.00.105 = 06.34.00.105 [vdf database (part 1), on-disk]
06.34.00.106 = 06.34.00.106 [vdf database (part 2), on-disk]
06.34.00.124 = 06.34.00.124 [vdf database (part 3), on-disk]
06.34.00.14 = 06.34.00.14 [scan engine, on-disk]

scan engine 06.33.00.11 --> 06.34.00.14 (/usr/lib/AntiVir/antivir)
vdf database 06.33.00.07 --> 06.34.00.124 (/usr/lib/AntiVir/antivir1.vdf,
/usr/lib/AntiVir/antivir2.vdf, /usr/lib/AntiVir/antivir3.vdf)

AntiVir updated successfully
```

If you only want to check whether new updates are available without updating the files, enter

**`/usr/lib/AntiVir/antivir --update --check`**

The output looks like the following:

```
AntiVir / Linux Version 2.1.5-24 +gui
Copyright (c) 1994-2005 by H+BEDV Datentechnik GmbH.
All rights reserved.

checking for updates

06.32.00.60  = 06.32.00.60  [vdf database (part 0), on-disk]
06.32.18.16  < 06.34.00.105 [vdf database (part 1), on-disk]
06.32.18.17  < 06.34.00.106 [vdf database (part 2), on-disk]
06.33.00.07  < 06.34.00.124 [vdf database (part 3), on-disk]
06.33.00.11  < 06.34.00.14  [scan engine, running]
06.33.00.11  < 06.34.00.14  [scan engine, on-disk]

an update for the scan engine is available (/usr/lib/AntiVir/antivir)
an update for the VDF database is available
(/usr/lib/AntiVir/antivir1.vdf, /usr/lib/AntiVir/antivir2.vdf,
/usr/lib/AntiVir/antivir3.vdf)
```

## **Exercise 12-1      Use AVMailGate as a Virus Scanner for Email**

In this exercise, you install and configure AVMailGate as a virus scanner for mails. Finally, you update the AVMailGate virus signatures.

Do the following:

- Part I - Install AVMailGate
- Part II - Configure Postfix to Use AVMailGate as Content Filter
- Part III - Configure the Ports for AVMailGate to Use
- Part IV - Check Configuration Using a Virus File from CD
- Part V - Update Your Virus Signatures

### **Part I - Install AVMailGate**

1. From the main menu, start **YaST**.
2. Enter the root password (**novell**) and select **OK**.
3. From the YaST Control Center, select **Software > Software Management**.
4. From the filter drop-down menu, select **Search**.
5. In the Search field, enter **avmailgate**; then select **Search**.
6. On the right, select the **avmailgate** package.
7. Select **Accept**; then insert the *SUSE Linux Enterprise Server 10* DVD.
8. Select **Continue** to resolve dependencies.
9. When installation is complete, remove the DVD and close the YaST Control Center.

## Part II - Configure Postfix to Use AVMailGate as Content Filter

1. Open the file `/etc/postfix/master.cf` in a text editor.
2. Uncomment the following line (on one line):  
**`localhost:10025 inet n - n - - smtpd -o content_filter=`**
3. Add the following line in `/etc/postfix/main.cf`:  
**`content_filter = smtp:127.0.0.1:10024`**
4. Save the file.
5. Enter **`postfix reload`**.

## Part III - Configure the Ports for AVMailGate to Use

1. To ensure that AVMailGate listens on port 10024 and not on port 25, edit `/etc/avmailgate.conf`:  
**`ListenAddress 127.0.0.1 port 10024`**
2. To ensure that AvMailGate sends mails back to Postfix via SMTP on host localhost via port 10025, edit `/etc/avmailgate.conf`:  
**`ForwardTo SMTP: localhost port 10025`**
3. Because AvMailGate sends out notification messages as AvMailGate, set an alias in `/etc/aliases`:  
**`avmailgate: root`**
4. Enter **`newaliases`**.
5. Enter **`rcavgate start`**.

## Part IV - Check Configuration Using a Virus File from CD

1. Log in as user `geeko`.
2. Send an infected mail to user `root` by entering  
**`mail root -s "Virus Test" -a /media/cdrecorder/section_4/sample-virus-executable.txt`**

3. Enter some text for the email message  
**This is an infected mail.**  
.
4. Log in as user root.
5. Check whether the mail queue is empty by entering  
**mailq**
6. Check whether the infected mail arrived by entering  
**mail**
7. Check whether the infected mail was detected by entering  
**ls /var/spool/avmailgate/rejected**

### **Part V - Update Your Virus Signatures**

1. To check for a new version of the virus signatures, enter  
**/usr/lib/AntiVir/antivir --update --check**
2. To download the virus signatures, enter  
**/usr/lib/AntiVir/antivir --update**

***(End of Exercise)***

## ***AMaViSd-new***

AMaViSd-new (*A Mail Virus Scanner*) consists of the daemon and some optional helper programs, which are only needed during setup between the message transfer agent (MTA) and one or more content checkers (virus scanners or SpamAssassin).

The mail server sends all incoming and outgoing mails to AMaViSd. The email will be extracted and tested by AMaViSd-new.

AMaViSd recognizes four kinds of unwanted mails. They are mails that have:

1. Invalid headers
2. Banned file types
3. Viruses
4. Spam

AMaViSd tests incoming mails for these four types in the order listed.

If nothing bad is found, AMaViSd-new will send the email back to the mail server, ready for delivery.

By default, AMaViSd-new listens at port 10024 for incoming email from the mail server.

AMaViSd-new sends clean mails to the mail server via port 10025. It is normally positioned at or near a central mail server, not necessarily where users' mailboxes and final delivery takes place.

This topic describes the following:

- Install AMaViSd-new
- Configure AMaViSd-new

## Install AMaViSd-new

To install AMaViSd-new, select the package **amavisd-new**.



---

AMaViSd-new will not work if no virus scanner is installed on your system. If you want to use AMaViSd-new only for spam filtering, search for **@bypass\_virus\_checks\_acl** in `/etc/amavisd.conf` and remove the comment sign (“#”) at the beginning of the line.

---

The compressing tools `gzip`, `bzip2`, `arc`, `lha`, `unrar`, `zoo`, `cpio`, and `lzop` should be installed, too.

During the SUSE Linux Enterprise Server 10 installation, a user `vscan` is created. It is used for AMaViSd.

```
da51:~ # grep vscan /etc/passwd
vscan:x:65:104:Vscan account:/var/spool/amavis:/bin/false
```

After installing AMaViSd-new, you can start the daemon with **rcamavis start**.

The daemon should listen on port 10024.

```
da51:~ # telnet 127.0.0.1 10024
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
220 [127.0.0.1] ESMTP amavisd-new service ready
```



Using SMTP commands you can write an email now.

```
da51:~ # telnet 127.0.0.1 10024
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
220 [127.0.0.1] ESMTP amavisd-new service ready
mail from: jgoldman@digitalairlines.com
250 2.1.0 Sender jgoldman@digitalairlines.com OK
rcpt to: geeko@digitalairlines.com
250 2.1.5 Recipient geeko@digitalairlines.com OK
data
354 End data with <CR><LF>.<CR><LF>
Subject: Test

Test
.
250 2.6.0 Ok, id=09232-01, from MTA([127.0.0.1]:10025): 250
Ok: queued as 5747B16A68
quit
221 2.0.0 [127.0.0.1] amavisd-new closing transmission
channel
Connection closed by foreign host.
da51:~ #
```

The email header should contain a line for amavisd.

```
geeko@da51:~> mail
mailx version nail 11.25 7/29/05.  Type ? for help.
"/var/spool/mail/geeko": 1 message 1 new
>N 1 jgoldman@digitalai Tue May 16 15:30 20/829 Test
? 1
Message 1:
From jgoldman@digitalairlines.com Tue May 16 15:30:58 2006
X-Original-To: geeko@digitalairlines.com
Delivered-To: geeko@digitalairlines.com
Subject: Test
X-Virus-Scanned: amavisd-new at example.com
Date: Tue, 16 May 2006 15:30:58 -0400 (EDT)
From: jgoldman@digitalairlines.com
To: undisclosed-recipients;;

Test

?
```

## Configure AMaViSd-new

On SUSE Linux Enterprise Server 10 you can configure AMaViSd-new by editing one of the following files:

- `/etc/sysconfig/amavis`
- `/etc/amavisd.conf`

### **`/etc/sysconfig/amavis`**

The configuration file `/etc/sysconfig/amavis` is only available on SUSE Linux products.

In this file there are only two parameters:

- **USE\_AMAVIS.** If set to **yes**, Sendmail or Postfix are prepared to use AMaViSd-new.

- **AMAVIS\_SENDMAIL\_MILTER.** If set to **yes**, the milter (*Mail Filter*) interface of Sendmail will be started.

Using the milter interface, it is possible to connect mail filter applications to Sendmail in a standardized form.

After changing the file `/etc/sysconfig/amavis`, you have to run the command **SuSEconfig**.

Setting `USE_AMAVIS` to `yes` makes three changes in `/etc/postfix/master.cf`:

- The `smtp` protocol

```
smtp      inet  n       -       n       -       2       smtpd
-o content_filter=smtp:[127.0.0.1]:10024
```

- The `smtps` protocol (still marked as comment)

```
#smtps    inet  n       -       n       -       2       smtpd
-o smtpd_tls_wrappermode=yes -o content_filter=smtp:[127.0.0.1]:10024
```

- Port 10025

```
localhost:10025 inet      n       -       n       -       -       smtpd
-o content_filter=
```

In this file you also have to add a process for `amavis`:

```
smtp-amavis unix  -       -       n       -       2       smtp
-o smtp_data_done_timeout=1200
-o smtp_send_xforward_command=yes
-o disable_dns_lookups=yes
-o max_use=20
```

In the Postfix configuration file `/etc/postfix/main.cf` you have to add the following line:

```
content_filter = smtp-amavis:[127.0.0.1]:10024
```

Restart Postfix by entering **rcpostfix reload**.

### **/etc/amavisd.conf**

AMaViSd is configured by editing the file `/etc/amavisd.conf`. The syntax of this configuration file is plain Perl code. Because of this, each line ends in a semicolon.

In Perl strings in double quotation marks can include variables that start with “\$” or “@”. To include characters “@” and “\$” in double-quoted strings, they must be preceded by a backslash.

In single-quoted strings, the “\$” and “@” lose their special meaning, so it is usually easier to use single quoted strings.

In both cases, the backslash needs to be doubled.

In the documentation directory of AMaViSd-new (`/usr/share/doc/packages/amavisd-new/`), two more variants of the configuration file are available:

- **amavisd.conf-default**. Includes all possible parameters with their defaults.
- **amavisd.conf-sample**. A more structured file with a lot of explanations and examples.

In this section we want to discuss some of the most important parameters in order of occurrence in the `/etc/amavisd.conf` file.

- **@bypass\_virus\_checks\_maps**. This array includes a list of virus lookup tables. You can disable virus checking by setting this array to “1” (uncomment the line).

- **@bypass\_spam\_checks\_maps.** This array includes a list of spam lookup tables. You can disable spam checking by setting this array to “1” (uncomment the line).



---

Next to @bypass\_virus\_checks\_maps and

@bypass\_spam\_checks\_maps, there are two more arrays of the same kind available:

@bypass\_banned\_checks\_maps enables checking for banned names or file types.

@bypass\_header\_checks\_maps enables checking for invalid headers.

---

- **\$max\_servers.** Number of pre-forked children.

Should match the number of your MTA pipe, e.g., the **maxproc** field in /etc/postfix/master.cf.

```
#
=====
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)    (yes)   (yes)   (never) (100)
#
=====
smtp      inet  n       -       n       -       2       smtpd -o
content_filter=smtp:[127.0.0.1]:10024
```

- **\$daemon\_user** and **\$daemon\_group.** User and group to which the daemon will change.
- **\$mydomain.** The domain name.
- **\$MYHOME.** Location where the AMaViSd-new files are stored.
- **\$TEMPBASE.** The path of a temporary directory that can be used by AMaViSd-new. This directory must exist or needs to be created manually.

In the following line an environment variable **\$TMPDIR** is defined with the content of the **\$TEMPBASE** variable.

- **\$QUARANTINEDIR**. The path to where infected mails can be put. This can be
  - **a file**. Path does not end with backslash.
  - **a directory**. Path ends with backslash.
  - **disabled**. Leave it empty.
- **\$quarantine\_subdir\_levels**. If set to “1” subdirectories in \$QUARANTINEDIR are created to disperse quarantine.
- **\$daemon\_chroot\_dir**. Run the daemon in the specified chroot jail. If you do not want chroot, leave it empty.
- **\$db\_home**. Path of the databases. Default: \$MYHOME/db.
- **\$helpers\_home**. Sets the environment variable \$HOME. The value is passed to other SpamAssassin modules.
- **\$pid\_file**. Path of the PID file.
- **\$lock\_file**. Path of the lock file.
- **@local\_domains\_maps**. List of lookup tables that can be used to decide whether a recipient is local or not, i.e., if the message is outgoing or not.
- **@mynetworks**. List of IP ranges which determines if the original SMTP client IP address belongs to the internal networks, i.e. if mail is coming from inside.
- **\$log\_level**. Log level.
  - **0**. Startup/exit/failure messages, viruses detected
  - **1**. Args passed from client, some more interesting messages
  - **2**. Virus scanner output, timing
  - **3**. Server, client
  - **4**. Decompose parts
  - **5**. More debug details

- **\$log\_recip\_tmpl.** Template for log file entries.  
A list of the available macros can be found in  
/usr/share/doc/packages/amavisd-new/README\_FILES/  
README.customize.
- **\$DO\_SYSLOG.** If set to “1”, the syslog daemon is used for  
loggings.
- **\$SYSLOG\_LEVEL.** Level of syslog loggings.
- **\$enable\_db.** Enable use of BerkeleyDB/libdb. If it is enabled,  
you can also enable the libdb cache using  
**\$enable\_global\_cache.**
- **\$inet\_socket\_port.** Port number that the AMaViSd-new should  
listen to.
- **\$unix\_socketname.** If you are using Sendmail milter, you have  
to enter the socket name here.
- **\$sa\_tag\_level\_deflt.** Spam info headers are added to the mail if  
the spam level is at or above the given number.
- **\$sa\_tag2\_level\_deflt.** Spam detected headers are added to the  
mail if the spam level is at or above the given number.
- **\$sa\_kill\_level\_deflt.** Spam evasive actions (bounce/reject/drop)  
are triggered if the spam level is at or above the given number.
- **\$sa\_dsn\_cutoff\_level.** Spam with a spam level beyond this  
number is not sent.
- **\$sa\_quarantine\_cutoff\_level.** Spam with a spam level beyond  
this number is not quarantined.
- **\$sa\_mail\_body\_size\_limit.** Email messages larger than the  
given number is not passed to SpamAssassin. (Less than 1% of  
spam is larger than 64 KB.)
- **\$sa\_local\_tests\_only.** If set to “1”, no SpamAssassin tests  
requiring Internet access are performed.

- **\$sa\_auto\_whitelist.** If set to “1”, AWL (auto-whitelist) in SpamAssassin 2.63 or older is turned on (irrelevant for SpamAssassin 3.0; on SUSE Linux Enterprise Server 10, version 3.1.0 is available)




---

AWL tracks scores for your regular correspondents in a small on-disk database. Since version 3.0, it is enabled by default.

---

- **@lookup\_sql\_dsn.** Array with information about where to find SQL server(s) and database to support SQL lookups. One item includes a triple of data: source name, user, and password.
- **\$virus\_admin.** Fully qualified address of the antivirus administrator.
- **\$mailfrom\_notify\_admin.** Fully qualified address of the sender of admin notifications.
- **\$mailfrom\_notify\_recip.** Fully qualified address of the sender of virus notifications.
- **\$mailfrom\_notify\_spamadmin.** Fully qualified address of the sender of spam notifications.
- **\$mailfrom\_to\_quarantine.** Whom quarantined messages appear to be sent from. If undefined, the original sender is used.
- **@addr\_extension\_virus\_maps.** The specified string is added to the recipient’s address if a virus is detected.

The **@addr\_extension\_spam\_maps**,  
**@addr\_extension\_banned\_maps**, and  
**@addr\_extension\_bad\_header\_maps** strings work in the same way.

The string is separated from the recipient address by the string specified in **\$recipient\_delimiter**.

Example:

geeko@digitalairlines.com > geeko+spam@digitalairlines.com



- **\$path**. The content of this variable is passed to the PATH environment variable.
- **\$dspam**. Activate the dspam content filter (<http://www.nuclearelephant.com/projects/dspam/>).
- **\$MAXLEVELS**. Maximum recursion level for extraction and decoding.
- **\$MAXFILES**. Maximum number of extracted files.
- **\$MIN\_EXPANSION\_QUOTA**,  
**\$MAX\_EXPANSION\_QUOTA**. Minimum and maximum storage size (in bytes) that is available for mail extraction.
- **\$sa\_spam\_subject\_tag**. String that is prepended to the subject header when message exceeds **\$sa\_tag2\_level\_deflt** level.
- **@\*\_lovers\_maps**. Email to the specified recipients is not examined and filtered.
- **@blacklist\_sender\_maps**. A message from a blacklisted envelope sender address is marked as spam.  
  
A **@whitelist\_sender\_maps** is also available. Mail with sender addresses from this array are delivered although they are marked as spam.
- **@score\_sender\_maps**. Using this variable you can add or subtract a specified value to/from the spam value.

The next variables (beginning with **@viruses\_that\_fake\_sender\_maps**) contain regular expressions to filter mails.

Many regular expressions are predefined and you can enable them by removing the comment hash sign at the beginning of the line(s).

Of course you can modify the regular expressions if they do not fit your needs.

Some other important variables are:

- **\$final\_virus\_destiny, \$final\_banned\_destiny, \$final\_spam\_destiny, \$final\_bad\_header\_destiny.** Defines what to do with email that has a virus, a banned sender, spam content, or incorrect headers.

You can use the following values:

- **D\_PASS.** Mail will pass to recipients, regardless of bad contents.
- **D\_DISCARD.** Mail will not be delivered to its recipients; sender will not be notified.
- **D\_BOUNCE.** Mail will not be delivered to its recipients; a nondelivery notification (bounce) will be sent to the sender by AMaViSd-new.
- **D\_REJECT.** Mail will not be delivered to its recipients; the sender should preferably get a rejection notification, (SMTP permanent reject response or nondelivery notification from the MTA).

If this is not possible, AMaViSd-new sends a bounce by itself (the same as **D\_BOUNCE**).

- **\$notify\_method.** Specify a host and port where the notifications are sent to.
- **\$notify\_sender\_tmpl.** Add this variable if you do not want the sender of an email notified.
- **\$notify\_virus\_sender\_tmpl.** Specify a text file if you do not want the default notification sent to the sender of an email that contained a virus.
- **\$notify\_virus\_admin\_tmpl.** Specify a text file if you do not want the administrator notified when a virus is detected.

- **\$notify\_virus\_recips\_tmpl.** Specify a text file if you do not want to notify the recipients of an email that contained a virus.
- **\$notify\_spam\_sender\_tmpl.** Specify a text file if you do not want to notify the sender of an email that contained spam.
- **\$notify\_spam\_admin\_tmpl.** Specify a text file if you do not want to notify the administrator if a spam email is detected.

## **Exercise 12-2      *Use AMaViSd as Virus Scanner for Email***

In this exercise, you install and configure AMaViSd. Virus notifications should be sent to user root. You test your configuration by using telnet and by sending a test virus file by mail.

Do the following:

- Part I - Install AMaViSd
- Part II - Change /etc/sysconfig/amavis
- Part III - Change /etc/amavisd.conf
- Part IV - Test the Configuration
- Part V - Send a Virus Email

### **Part I - Install AMaViSd**

1. From the main menu, start **YaST**.
2. Enter the root password (**novell**) and select **OK**.
3. From the YaST Control Center, select **Software > Software Management**.
4. From the filter drop-down menu, select **Search**.
5. In the Search field, enter **amavis**; then select **Search**.
6. On the right, select the **amavisd-new** package.
7. Select **Accept**; then insert the *SUSE Linux Enterprise Server 10* DVD.
8. Select **Continue** to resolve dependencies.
9. When installation is complete, remove the DVD and close the YaST Control Center.

## Part II - Change /etc/sysconfig/amavis

1. Open the file /etc/sysconfig/amavis by entering  
**vi /etc/sysconfig/amavis**
2. Change the line with the variable USE\_AMAVIS to  
**USE\_AMAVIS="yes"**
3. Exit vi by entering **:wq**.
4. Enter **SuSEconfig**.
5. Look at the messages of the output. If the file /etc/postfix/master.cf is left untouched, overwrite this file by entering  
**mv /etc/postfix/master.cf.SuSEconfig /etc/postfix/master.cf**
6. Open the file /etc/postfix/master.cf by entering  
**vi /etc/postfix/master.cf**
7. Add the following lines to the file /etc/postfix/master.cf:  
**smtp-amavis unix - - n - 2 smtp**  
**-o smtp\_data\_done\_timeout=1200**  
**-o smtp\_send\_xforward\_command=yes**  
**-o disable\_dns\_lookups=yes**  
**-o max\_use=20**
8. Exit vi by entering **:wq**.
9. Open the file /etc/postfix/main.cf by entering  
**vi /etc/postfix/main.cf**
10. To remove Procmail from the mailbox\_command (entered in a previous exercise), enter:  
**mailbox\_command =**
11. Add the following line to /etc/postfix/main.cf:  
**content\_filter = smtp-amavis:[127.0.0.1]:10024**
12. Exit vi by entering **:wq**.
13. Restart Postfix by entering **rcpostfix reload**.

### Part III - Change /etc/amavisd.conf

1. Open the file /etc/amavis.conf by entering  
**vi /etc/amavis.conf**
2. Modify the \$mydomain variable to  
**\$mydomain = 'digitalairlines.com';**
3. Change the mail address where virus notifications should be sent to root:  
**\$virus\_admin = "root\@\$mydomain";**
4. Exit vi by entering **:wq**.
5. Start the AMaViSd by entering **rcamavis start**.

### Part IV - Test the Configuration

1. Check whether Postfix listens on port 10025 by entering  
**telnet 127.0.0.1 10025**
2. Enter **quit**.
3. Check whether the AMaViSd listens on port 10024 by entering  
**telnet 127.0.0.1 10024**
4. Enter **mail from: jgoldman@digitalairlines.com**
5. Enter **rcpt to: geeko@digitalairlines.com**
6. Enter **data**
7. Open the file  
/usr/share/doc/packages/amavisd-new/test-messages/  
sample-virus-simple.txt and copy the last line into the clipboard.  
The line looks like this:  
**X5O!P% @AP[4PZX54(P^)7CC)7}\$EICAR-STANDARD-  
ANTIVIRUS-TEST-FILE!\$H+H\***

8. Paste the content of the clipboard into the first terminal and do the following:
  - a. Press **Enter**.
  - b. Type **.** (dot).
  - c. Press **Enter**.
9. You should get a virus warning like the following one:  
**250 2.7.1 Ok, discarded, id=14069-01-2 - VIRUS:  
Eicar-Test-Signature**
10. Enter **quit**.

### **Part V - Send a Virus Email**

1. Log in as user jgoldman by entering **su - jgoldman**.
2. Send a virus mail to user tux by entering  
**mail geeko@digitalairlines.com <  
/usr/share/doc/packages/amavisd-new/test-messages/sample  
-virus-simple.txt**
3. Log out by entering **exit**.
4. As root, enter **mail** to look for new email.

There should be an email from **virusalert** in your mail folder.

***(End of Exercise)***

## Summary

Objective	Summary
1. Understand SMTP Communication	<p>The following SMTP commands must be implemented in an SMTP server to provide SMTP communication:</p> <ul style="list-style-type: none"><li>■ <b>HELO</b></li><li>■ <b>MAIL FROM:</b></li><li>■ <b>RCPT TO:</b></li><li>■ <b>DATA</b></li><li>■ <b>RSET</b></li><li>■ <b>NOOP</b></li><li>■ <b>QUIT</b></li></ul>
2. Manage Spam	<p><b>spamassassin</b> (and <b>spamc</b>) expect their input from STDIN.</p> <p>If you want to use SpamAssassin with your Postfix configuration, the easiest way is to use Procmail.</p> <p>Spam can be filtered from the mail client or in a further Procmail configuration (<code>~/procmailrc</code>).</p> <p>You can use <b>telnet</b> to test your configuration. You also can send a spam email directly using the <b>sendmail</b> command.</p>



Objective	Summary
3. Use a Virus Scanner for Email	<p>AVMailGate is an antivirus email filter from H+BEDV Datentechnik GmbH.</p> <p>AVMailGate can update the virus definition file and the engine itself.</p> <p>AVMailGate is composed of two kinds of processes:</p> <ul style="list-style-type: none"><li>■ <b>avgated.</b> The smtpd receiver that stores incoming email in the input queue.</li><li>■ <b>avgatefwd.</b> Virus scanning function and SMTP forwarder and sendmail invoker.</li></ul> <p>Both processes can be configured by editing the file <b>/etc/avmailgate.conf</b>.</p> <p>AMaViSd-new consists of</p> <ul style="list-style-type: none"><li>■ The daemon.</li><li>■ Optional helper programs that are only needed in setup between the message transfer agent (MTA).</li><li>■ One or more content checkers: virus scanners such as SpamAssassin.</li></ul> <p>The email server sends all incoming and outgoing email to AMaViSd.</p> <p>The emails will be extracted by AMaViSd-new and tested with a virus scanner.</p>

---

Objective	Summary
3. Use a Virus Scanner for Email (continued)	<p>If no viruses are found, AMaViSd-new sends the email back to the mail server ready for delivery.</p> <p>AMaViSd-new is configured in the file <b>/etc/amavisd.conf</b>.</p>

## SECTION 13 Apply Security

There are two important changes concerning security:

- The Novell Customer Center manage your business and technical interactions with Novell. This is also important for using the YaST Online update.
- The syslog-ng used to log system messages on SUSE Linux Enterprise Server 10.

### Objectives

1. Apply Security Updates
2. Understand Recent Match of iptables
3. Log to a Remote Host

## **Objective 1      Apply Security Updates**

SUSE Linux Enterprise Server 10 is sold with system maintenance. This system maintenance includes updates and security patches.

Software updates can be managed with the YaST Online Update (YOU) module. This YaST module downloads and installs software updates and security patches.

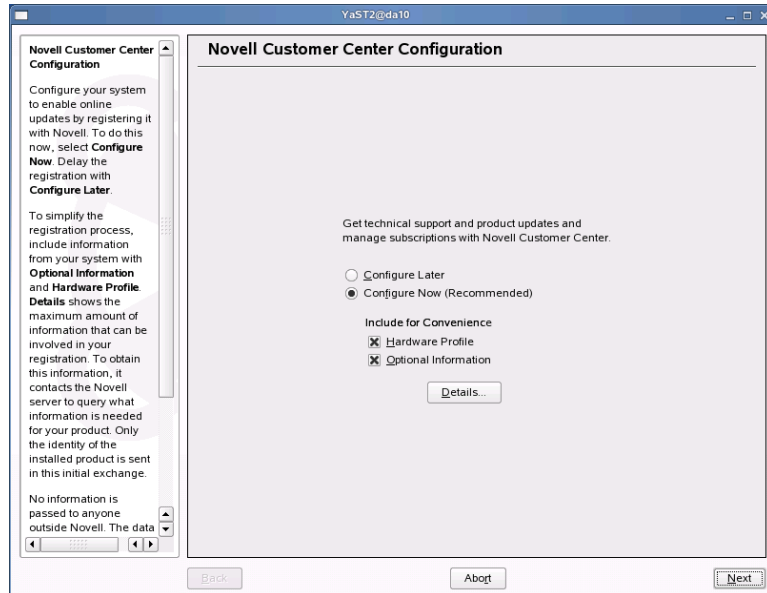
To apply security updates, you need to do the following:

- Configure the Novell Customer Center
- Use the YaST Online Update

## Configure the Novell Customer Center

To access the configuration of the Novell Customer Center, start the YaST Control Center and select **Software > Novell Customer Center Configuration**. You can also start the dialog directly from a terminal window as root by entering `yast2 inst_suse_register`. The dialog is the same as that offered during installation for this purpose:

Figure 13-1



Selecting Details shows what information is being collected and sent.

With a browser, the Novell Customer Center can be accessed at <http://www.novell.com/center/>. After you have created a Novell account, you need to register your product with the registration code delivered with the SUSE Linux Enterprise Server 10 product.



The SUSE Linux Enterprise Server 10 DVD you received as part of your student kit does not include maintenance. Visit the Novell Customer Center for information on how to take part in the SUSE Linux Enterprise Server 10 Maintenance Program.

Only registered products with a valid maintenance contract can be updated with the YOU module.

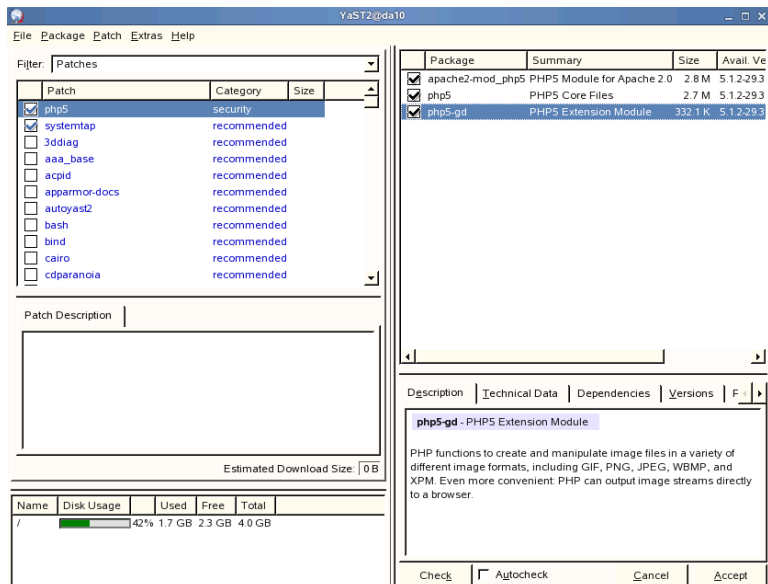
## Use the YaST Online Update

The following is a quick guide to applying software updates with YOU.

Start the YOU module from the YaST Control Center by selecting **Software > Online Update**.

The following appears:

**Figure 13-2**

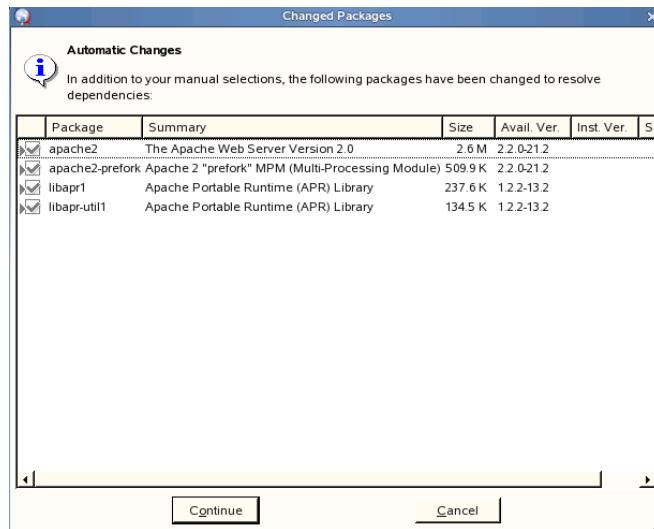


On the top left side of the dialog all available patches are displayed. Select an entry to see details for the update on the right side of the dialog. To have an update installed in the next step, select the check box next to the corresponding entry.

Select **Accept** to start the update process.

Depending on your selection an **Automatic Changes** dialog appears:

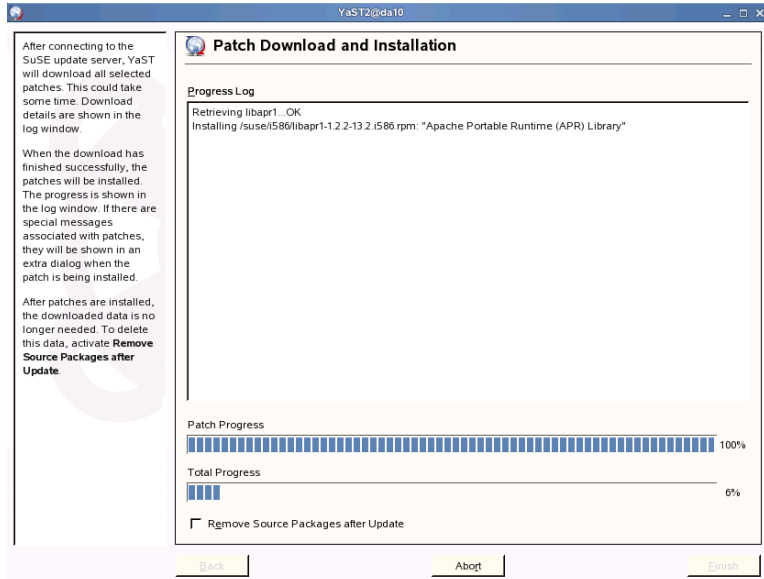
**Figure 13-3**



Accept the changes by selecting **Continue**; the patches are transferred from the update server and installed.

During the installation process YOU displays the following dialog:

**Figure 13-4**



Sometimes additional information is displayed for some updates. These dialogs need to be confirmed to install the corresponding software package.

Once all updates have been downloaded and installed, select **Finish** to close the dialog.



## Objective 2      Understand Recent Match of iptables

The recent match can be used to dynamically create a list of IP addresses and create rules regarding those IP addresses.

There are, for instance, automated attacks that try to guess accounts and passwords via ssh. Using the recent match, you can temporarily block traffic from the machine that originates this attack.

The information is kept in `/proc/net/ipt_recent/DEFAULT`, unless you specify a different name with the option `--name`. Specifying a name is useful if you want to keep track of different IP addresses in various rules.

The following is an example for ssh:

```
# Drop if 2 hits or more within the last 60 seconds
iptables -A INPUT -p tcp --dport 22 -i eth0 -m state --state NEW -m recent
--update --seconds 60 --hitcount 2 -j DROP
# Add the source address to the list
iptables -A INPUT -p tcp --dport 22 -i eth0 -m state --state NEW -m recent
--set
```

For the first ssh connection, the first rule does not match. The second rule creates a new entry in the recent list `DEFAULT`; because there is no target defined with `-j`, the packet travels down the chain and gets checked by subsequent rules.

For the second ssh connection the hitcount is one, so the first rule still does not match. The second rule updates the entry in the `DEFAULT` list, and the hitcount is now 2.

When the third packet arrives within a minute after the second packet, the first rule matches and the packet is dropped.

A legitimate user who mistyped his password would have to wait 60 seconds to have another try after two failures.

## Objective 3      Log to a Remote Host

The syslog-ng used to log system messages on SUSE Linux Enterprise Server 10 can send messages to and receive messages from other computers. The receiving computer is sometimes referred to as loghost.

Configuration changes are necessary on both sides:

- Client Side Configuration of Syslog-ng
- Server Side Configuration of Syslog-ng

### ***Client Side Configuration of Syslog-ng***

Logging to other computers is configured in **/etc/syslog-ng/syslog-ng.conf.in**, as in the following example:

```
destination tologhost { udp(10.0.0.254 port(514)); };  
log { source(src); destination(tologhost); };
```

The first line defines a destination ***tologhost***, with protocol, IP address (you could also use the host name), and port. The second line configures logging to this destination. Because no filter is included, all messages from the source ***src*** get sent to the destination ***tologhost***.

After editing the file **/etc/syslog-ng/syslog-ng.conf.in**, run **SuSEconfig --module syslog-ng** to transfer the changes to **/etc/syslog-ng/syslog-ng.conf**, which is the actual configuration file used by syslog-ng.

## ***Server Side Configuration of Syslog-ng***

To receive messages, syslog-ng has to listen on a port, usually port 514. Unlike syslogd, which only supports UDP connections, syslog-ng also supports TCP.

To bind to a port, you have to add a line within a source section. You can either add it to the existing **source src** section, or create a new one, as in the following:

```
source network {  
    udp(ip("10.0.0.254") port(514));  
};
```

Using 0.0.0.0 as the IP address causes syslog-ng to bind to all available interfaces.

Then a destination and a log entry are required:

```
destination digiair { file("/var/log/$HOST"); };  
log { source(network); destination(digiair); };
```

The first line defines the destination digiair; each host's log entries get written to a file with the hostname as file name. The log entry directs messages from the source network to the destination digiair.

### **Exercise 13-1      Log to a Remote Host**

In this exercise you work with a partner. Decide who will send messages and who will receive them. (If both of you send and receive messages to each other, you might create an endless loop.)

On the loghost configure syslog-ng to receive log messages from other hosts.

On the client configure syslog-ng to send log messages to the loghost.

#### **Detailed Steps to Complete this Exercise**

- Part I: On the Computer Receiving Messages
- Part II: On the Computer Sending Messages

#### **Part I: On the Computer Receiving Messages**

Do the following:

1. Open a terminal window and **su -** to root (password **novell**).
2. Open the file **/etc/syslog-ng/syslog-ng.conf.in** in vi.

Add another source section after the source **src** section, as in the following:

```
source network {  
    udp(ip("0.0.0.0") port(514));  
};
```

3. At the end of the file, add a destination and a log entry to log messages that arrive via the network:

```
destination digiair { file("/var/log/$HOST"); };  
log { source(network); destination(digiair); };
```

4. Save the file and quit vi.

5. Run SuSEconfig:

**SuSEconfig --module syslog-ng**

6. View /var/log/messages by entering

**tail -f /var/log/messages**

You should see a message that syslog-ng initialized its new configuration.

7. Quit tail by pressing **ctrl+c**.

8. Once messages are sent from the client, syslog-ng will create a file **/var/log/daxx**, for instance, /var/log/da20. View this file by entering:

**tail -F /var/log/daxx**

(Replace **daxx** by the hostname of your neighbor's computer.)

## Part II: On the Computer Sending Messages

Do the following:

1. Open a terminal window and **su -** to root (password **novell**).
2. Open the file **/etc/syslog-ng/syslog-ng.conf.in** in vi.

At the end of the file, add a destination and a log entry to send messages to the loghost:

```
destination tologhost { udp("daxx" port(514)); };  
log { source(src); destination(tologhost); };
```

Replace **daxx** in the above example with the host name of the loghost.

3. Save the file and quit vi.
4. Run SuSEconfig:

**SuSEconfig --module syslog-ng**

SuSEconfig causes syslog-ng to reread its configuration, so there is no need to do this in an extra step.

5. Create log entries by using **su -** in a terminal window or by using the program logger. The entries should appear on the loghost.

***(End of Exercise)***

## Summary

Objective	Summary
1. Apply Security Updates	<p>To get and apply security updates for SUSE Linux Enterprise Server 10, you need to do the following:</p> <ul style="list-style-type: none"><li>■ Register SLES 10 at <a href="http://www.novell.com/center/">http://www.novell.com/center/</a></li><li>■ Download and apply updates with YOU.</li></ul>
2. Understand Recent Match of iptables	<p>The recent match can be used to dynamically create a list of IP addresses and create rules regarding those IP addresses.</p> <p>The information is kept in <code>/proc/net/ipt_recent/DEFAULT</code>, unless you specify a different name.</p>
3. Log to a Remote Host	<p>Logging to a log host prevents manipulation of log files after an intrusion.</p> <p>The information in log files is only useful if someone sees it and acts accordingly.</p> <p>Several programs, like logcheck, logsurfer, or custom scripts exist to help find relevant log entries.</p>





## SECTION 14 AppArmor

Novell AppArmor is a mandatory access control scheme that lets you specify on a per program basis which files the program may read, write, and execute.

AppArmor secures applications by enforcing good application behavior without relying on attack signatures, so it can prevent attacks even if the attacks are exploiting previously unknown vulnerabilities.

### Objectives

1. Improve Application Security with AppArmor
2. Create and Manage AppArmor Profiles
3. Control AppArmor
4. Monitor AppArmor

## Objective 1    **Improve Application Security with AppArmor**

While you can keep your software up to date, you can still be hit with an attack that exploits a vulnerability that is not yet known or for which there is no fix yet.

The idea of AppArmor is to have the kernel limit what a software program can do. In addition to the limitations set by the usual user and group permissions, limitations are imposed based on a set of rules for a specific program.

Often an intruder does not directly gain root privileges. By exploiting a vulnerability in, for instance, a web server, she might be able to start a shell as the user running the web server. With that unprivileged access, she exploits yet another vulnerability in some other software program to gain root access.

With AppArmor, even if an intruder manages to find a vulnerability in some server software, the damage is limited due to the fact that the intruder may not access any files or execute any programs beyond what the application is allowed by AppArmor's rule set to access or execute in routine operation.

This concept is not limited to normal accounts; it also limits to a certain extent what an application running with root privileges may do. A confined process cannot call certain system calls, even if running as root. Thus even if an attacker gained root privileges, she would still be limited in what she might be able to do.

The AppArmor kernel modules (apparmor and aamatch\_pcre) hook into the Linux Security Modules Framework of the kernel.

Profiles in **/etc/apparmor.d/** are used to configure which application may access and execute which files.

AppArmor has to be activated before the applications it controls are started. Applications already running when AppArmor is activated are not controlled by AppArmor, even if a profile has been created for them. Therefore, AppArmor is activated early in the boot process by **/etc/init.d/boot.apparmor**.

In a default installation of SUSE Linux Enterprise Server 10, AppArmor is actively protecting several services using a set of profiles provided by Novell. Using the provided YaST modules or command line tools, you can easily adapt these to your needs and create new profiles for additional applications you want to protect.

As a general rule, you should confine programs that grant privilege, i.e., programs that have access to resources that the person using the program does not have:

- Network agents—programs that have open network ports
- Cron jobs
- Web applications



---

At some points in the documentation of AppArmor you will find AppArmor's former name, Subdomain.

---

## Objective 2      Create and Manage AppArmor Profiles

SUSE Linux Enterprise Server 10 comes with AppArmor Profiles for various applications, such as `named`, `ntpd`, `nsd`, and others.

The profiles are contained in `/etc/apparmor.d/`. The filename of the profile represents the filename of the application including the path, with “/” being replaced by a “.”: The profile for `/usr/sbin/squid` would be contained in `/etc/apparmor.d/usr.sbin.squid`.

A profile can include other files with an **#include** statement. The directory `/etc/apparmor/abstractions/` contains several files that are intended to be included in profiles, depending on the kind of program to be protected by AppArmor. There are abstractions for files that should be readable or writable by all programs (`base`), for nameservice-related files like `/etc/passwd` (`nameservice`), for files related to console operations (`console`), and others.

The profiles are plain text files and it is therefore possible to create and modify them with any text editor. However, command line tools as well as a YaST module greatly simplify the process of creating profiles.

In addition to the active profiles in `/etc/apparmor.d/`, several profiles are already prepared in `/etc/apparmor/profiles/extras/` that you can customize to your needs and copy to `/etc/apparmor.d/` to activate them.

To successfully administer AppArmor, you need to

- Understand Profiles and Rules
- Administer AppArmor Profiles with YaST
- Administer AppArmor Profiles with Command Line Tools

## ***Understand Profiles and Rules***

Novell AppArmor **profiles** contain two types of AppArmor **rules**: path entries and capability entries. Path entries specify what a process can access in the file system. AppArmor, by default, limits the capabilities a process is given (see man apparmor). Capability entries are used to specify specific POSIX capabilities (man 7 capabilities) a process is granted, overriding the default limitation.

Other files containing AppArmor rules can be pulled in with **#include** statements.

As an example, let's have a look at the profile for /sbin/klogd, the kernel log daemon (/etc/apparmor.d/sbin.klogd):

```
1 # Profile for /sbin/klogd
   #include <tunables/global>
5 /sbin/klogd {
   #include <abstractions/base>
   capability sys_admin,
10 /boot/System.map*      r,
   /proc/kmsg             r,
   /sbin/klogd            rmix,
   /var/log/boot.msg      rwl,
   /var/run/klogd.pid     rwl,
15 }
```

Comments (as in line1) start with a # sign,

**#include** (as in line 3 and 6) is not interpreted as a comment, but is used to include rules from other files. The path as given above is relative to **/etc/apparmor.d/**.

**/etc/apparmor.d/tunables/global** (line 3) is used to include definitions that should be available in every profile. By default, it just includes **/etc/apparmor.d/tunables/home**, which defines the variables **@{HOMEDIRS}** and **@{HOME}**. These variables are used in various profiles.

The directory **/etc/apparmor.d/abstractions/** contains files with general rules grouped by common application tasks. These include, for instance, access to files all applications need (base), access to authentication mechanisms (authentication), graphics environments (kde, gnome), name resolution (nameservice), and others. Instead of having these redundantly specified in several profiles, they are defined at one point and included in the profiles that need them.

Line 5 in the example above gives the absolute path to the program confined by AppArmor. The rules as well as any includes follow within the curly braces {}.

Line 8 enables the capability `sys_admin` for this program. Any other capabilities needed would be listed in separate lines starting with **capability**.

The remaining lines list files and directories, and the access permission granted.

Within lines listing files and directories, the following wildcards can be used:

- **\***. Substitutes any number of characters, except `/`.
- **\*\***. Substitutes any number of characters, including `/`. Use **\*\*** to include subdirectories.
- **?**. Substitutes any single character, except `/`.
- **[abc]**. Substitutes a, b, or c.
- **[a-d]**. Substitutes a, b, c, or d.
- **{ab,cd}**. Substitutes either ab or cd.

The permissions granted can be

- **r**. Allows the program to have read access to the resource. Read access is required for scripts, and an executing process needs this permission to allow it to dump core or to be attached to with `ptrace`.

- **w.** Allows the program to have write access to the resource. Files must have this permission if they are to be unlinked (removed).
- **l.** Link mode mediates access to symlinks and hardlinks and grants the privilege to unlink (remove) files.
- **m.** Allow executable mapping. This mode allows a file to be mapped into memory using mmap(2)'s PROT\_EXEC flag. This flag marks the pages executable; it is used on some architectures to provide non-executable data pages, which can complicate exploit attempts. AppArmor uses this mode to limit which files a well-behaved program (or all programs on architectures that enforce non-executable memory access controls) may use as libraries, to limit the effect of invalid -L flags given to ld(1) and LD\_PRELOAD, LD\_LIBRARY\_PATH, given to ld.so(8).
- **ix.** Inherit Execute Mode. The executed resource inherits the current profile.
- **px.** Discrete Profile Execute Mode. This mode requires that a profile be defined for the resource executed. If there is no profile defined, access is denied.
- **Px.** Discrete Profile execute mode -- scrub the environment. **Px** allows the named program to run in **px** mode, but AppArmor will invoke the Linux Kernel's unsafe\_exec routines to scrub the environment, similar to setuid programs. (See man 8 ld.so for some information on setuid/setgid environment scrubbing.)
- **ux.** Unconstrained Execute Mode. Allows the program to execute the resource without any Novell AppArmor profile being applied to the executed resource. ***This should only be used in rare exceptions.***
- **Ux.** Unconstrained execute -- scrub the environment. As ux, it should only be used in rare exceptions.

The last five, ix, px, Px, ux, and Ux, cannot be combined.

The manual page covering the syntax of the profiles is  
**man 5 apparmor.d**

## ***Administer AppArmor Profiles with YaST***

The profile for `/sbin/klogd`, given in the example above, is a rather short profile. When you browse through the profiles in `/etc/apparmor.d/` or `/etc/apparmor/profiles/extras/` you will see that these profiles can be much more complex.

AppArmor comes with several tools that help to create and maintain AppArmor profiles. YaST modules exist that provide a graphical interface to those tools.

You can accomplish various tasks with YaST:

- Create a New Profile
- Update a Profile
- Delete a Profile

### **Create a New Profile**

To create a new profile, you can

- Use the New Profile Wizard
- Create a New Profile Manually

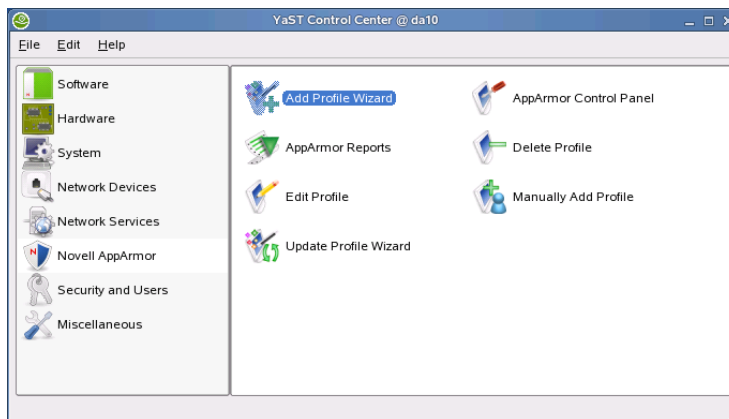
### **Use the New Profile Wizard**

There is a Wizard to create a new profile. Before calling the Wizard to profile an application, the first step is to stop the application you want to create a profile for.



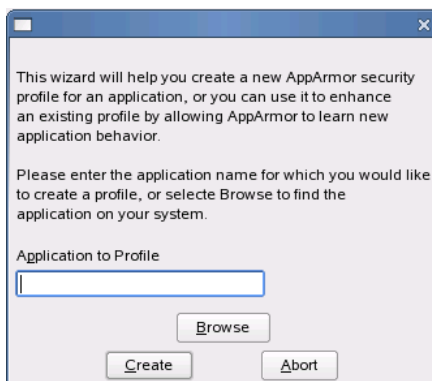
To access the Add Profile Wizard, start YaST and select Novell AppArmor:

**Figure 14-1**



Then select the Add Profile Wizard. The first step is to enter the application you want to profile:

**Figure 14-2**



If no path is given, the Wizard looks for the binary in the search path (variable \$PATH).

The next dialog asks you to start and use the application you want to profile. Use the application in a way that you expect it to be used in production. For instance, if you are profiling a web server, access it in a way that you expect it to be accessed during normal operation. For a web browser, use it in a way you expect the users to access web content.

During this learning phase, any access to files or capabilities needed by the application is granted as well as logged in the log file **/var/log/audit/audit.log**. Because any access is granted, you have to make sure that no attack can happen during this phase of profile creation. AppArmor does not yet protect your application.

Once you feel you have gone through all expected uses of the application, select **Scan system log for AppArmor events** in the YaST AppArmor Profile Wizard dialog.

For each event you are presented with a dialog where you can decide what should happen when this event occurs in the future. The dialog offers different options, depending on the event.

In case of access to a program, the dialog looks similar to the following:

Figure 14-3



- **Inherit.** The executed resource inherits the current (parent's) profile.
- **Profile.** Requires that a specific profile exist for the executed program.
- **Unconfined.** Executes the program without a security profile. Do not run unconfined unless absolutely necessary.
- **Deny.** The execution of the program will be denied.

In case of file access, the dialog offers different options:

**Figure 14-4**



The Add Profile Wizard suggests an access mode (r, w, l, or a combination of them). If more than one item appears in the list of files, directories, or #includes, select the radio button in front of the appropriate one, and then select one of these buttons:

- **Allow.** Grants the program access to the specified directory path.
- **Deny.** Prevents the program from accessing the specified file or directory.

Sometimes the suggested files or directories do not fit your needs. In this case, you can modify them:

- **Glob.** Selecting Glob once replaces the filename with an asterisk, including all files in the directory. Selecting Glob twice replaces the file and the directory it resides in by \*\*, including all directories and files of that level in the directory tree. Selecting Glob again goes up one level in the path.

- **Glob w/Ext.** Selecting Glob w/Ext once replaces the filename with an \*, but retains the file name extension: text.txt becomes \*.txt. Selecting Glob w/Ext twice replaces the file and the directory it resides in by \*\*, retaining the file name extension: /a/b/c/text.txt becomes /a/b/\*\*/\*.txt.
- **Edit.** Enables editing of the highlighted line. The new (edited) line appears at the bottom of the list.

After you have modified the line, select Allow or Deny. Go through each learning mode entry in this way.

Once all entries have been processed, you are returned to the AppArmor Profile Wizard dialog that asked you to run the application. You can run the application again, and then run through any additional entries generated by this.

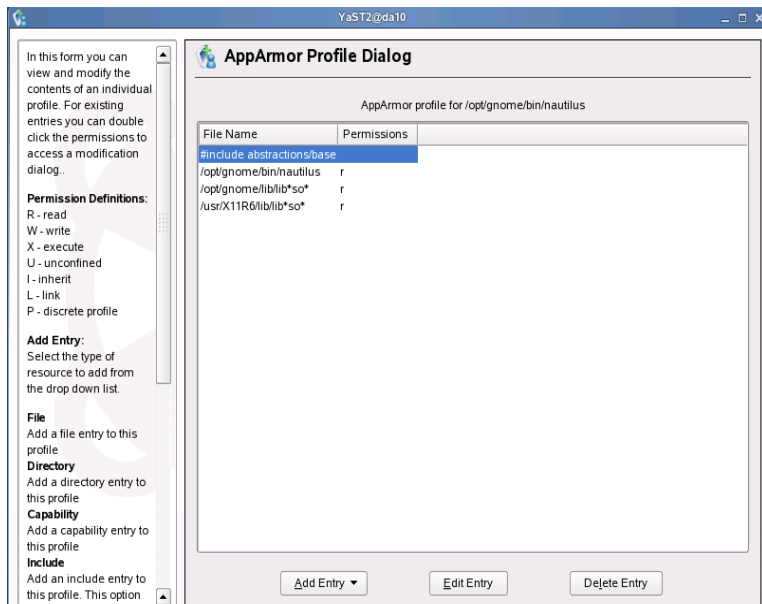
Once you conclude you are done, select **Finish**. The profile is written and activated.

If you want to discard your selections, select **Abort**.

## Create a New Profile Manually

When selecting **YaST > Novell AppArmor > Manually Add Profile**, you are prompted to select a file for which you want to create a profile. Subsequently the AppArmor Profile Dialog opens. There you can Add, Edit, and Delete entries to the profile by selecting the respective buttons.

**Figure 14-5**



The advantage of the YaST module is its syntax check. However, you can also use any text editor, like vi, to create and edit profiles.

## Update a Profile

To update a profile, you have two choices:

- Run the Add Profile Wizard Again
- Run the Update Profile Wizard

## Run the Add Profile Wizard Again

When you run the Add Profile Wizard on a program for which a profile already exists, the profiling does not start from scratch, but uses the existing profile as a basis.

You go through the same steps as if generating a new profile, but most likely you will have to answer fewer questions than on the first run.

This method is suitable to update a specific profile, especially for client applications that run a finite amount of time.

## Run the Update Profile Wizard

When you want to update several profiles, or profiles for applications that run over a longer period of time, using the Update Profile Wizard is the better choice.

Even though the Update Profile Wizard is a YaST module, you may need to take some preparatory steps with command line tools.

The first step is to decide which application profiles you want to update, and to put AppArmor into complain (also called learning) mode with regard to these applications. (If there is no profile yet for an application you want to profile, you have to create one first, using **autodep program**.)

The command **complain** is used to activate learning or complain mode. You can either use the program or the profile as the argument: For instance, both, **complain firefox** and **complain /etc/apparmor.d/usr.lib.firefox.firefox.sh** work to change AppArmor to learning mode for Firefox.

If you want to turn on learning mode for all applications confined by AppArmor, use **complain /etc/apparmor.d/\***.

In profiles that are in complain mode, the path to the application being confined is followed by **flags=(complain)**:

```
# Profile for /sbin/klogd

#include <tunables/global>

/sbin/klogd flags=(complain){
...
}
```

Then actually use your application(s) to create events in the log file.

The next step is to start the Update Profile Wizard by starting YaST and selecting **Novell AppArmor > Update Profile Wizard**:

**Figure 14-6**



The interface is almost identical to the Add Profile Wizard interface; also the choices you are presented do not differ.



However, as you are updating different profiles, you have to pay special attention to the profile in the first line to be sure that your decision on allowing or denying fits the respective profile.

Once the log file has been processed, select **Finish**. The profiles will be reloaded, but AppArmor is still in complain mode.

To have AppArmor again enforce the rules, use the command `enforce`, which has the same syntax as `complain`:  
**`enforce /etc/apparmor.d/*`** puts all profiles in enforce mode.

## Delete a Profile

To delete a profile, start YaST and select **Novell AppArmor > Delete Profile**. Select the profile to delete; then select Next. After you select **Yes** in the confirmation dialog, the profile is deleted and the application is no longer confined by AppArmor.

## ***Administer AppArmor Profiles with Command Line Tools***

There are various tools to create and maintain AppArmor profiles. These are

- `autodep`
- `genprof`
- `logprof`
- `vim`

### **autodep**

`autodep` generates a profile skeleton for a program and loads into the Novell AppArmor module in complain mode.

The syntax is **`autodep program1 program2 ...`**

## genprof

genprof (Generate Profile) is used to create a profile for an application. Stop the application you want to create a profile for before running genprof.

genprof runs autodep on the specified program if there is no profile yet, puts the new or already existing profile in complain mode, marks the log file, and prompts the user to start the program to profile and to exercise its functionality.

```
da10:~ # genprof firefox
Setting /usr/lib/firefox/firefox.sh to complain mode.

Please start the application to be profiled in
another window and exercise its functionality now.

Once completed, select the "Scan" button below in
order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the
opportunity to choose whether the access should be
allowed or denied.

Profiling: /usr/lib/firefox/firefox.sh

[(S)can system log for SubDomain events] / (F)inish
```

Once the user has done that and presses s in the terminal window where genprof is running, genprof calls logprof to run against the system log from where it was marked when genprof was started.

In case of access to a program, the dialog looks similar to the following:

```
Reading log entries from /var/log/audit/audit.log.
Updating subdomain profiles in /etc/apparmor.d.

Profile:   /usr/lib/firefox/firefox.sh
Program:   firefox
Execute:   /bin/basename
Severity:   unknown

[(I)nherit] / (P)rofile / (U)nconfined / (D)eny / Abo(r)t /
(F)inish
```

In case of access to a file or directory, the dialog looks similar to this:

```
Complain-mode changes:

Profile:   /usr/lib/firefox/firefox.sh
Path:      /dev/tty
Mode:      rw
Severity:   9

  1 - #include <abstractions/consoles>
  [2 - /dev/tty]

[(A)llow] / (D)eny / (G)lob / Glob w/(E)xt / (N)ew /
Abo(r)t / (F)inish
```

Press the number to switch lines as applicable, then press the letter in parenthesis corresponding to what you want to do. The options offered are the same as those within YaST; **New** corresponds to a certain extent to **edit** in YaST.

Once all log entries have been processed, you are returned to `genprof`, where you can start a new scan or Finish the profile generation.

```
Writing updated profile for /usr/lib/firefox/firefox.sh.  
Profiling: /usr/lib/firefox/firefox.sh  
[(S)can system log for SubDomain events] / (F)inish
```

## logprof

`logprof` is a tool used to scan the log `/var/log/audit/audit.log` for entries created by AppArmor for profiles in learning mode, and to interactively create new profile entries.

The choices you have are the same as those described under “`genprof`” on page 14-18.

If you want `logprof` to start scanning from a certain point in the log file, you can pass a string that describes that point. The following is an entry in the log file:

```
type=APPARMOR msg=audit(852099290.789:1103): PERMITTING w  
access to /root/.mozilla/firefox/uwgzf7zy.default/prefs.js  
(firefox-bin(8770) profile /usr/lib/firefox/firefox.sh  
active /usr/lib/firefox/firefox.sh)
```

Using the option `-m`: **`logprof -m "852099290.789:1103"`**, you can start the scan of the log file from that point, ignoring earlier entries.

## vim

The profiles can be changed using any text editor.

Compared to other editors, vim has the advantage that AppArmor includes a syntax highlighting description that enables vim to highlight syntax elements in profiles.



When confining Apache2, you can create subprofiles (also called hats) that use a different security context, for instance for pages using mod\_php5. To make use of such subprofiles, an application has to be made “hat-aware.” For Apache2 this is achieved with module mod\_change\_hat that comes with AppArmor on SLES10. For details on hats, see **man change\_hat** and section 5 of the *Administration Guide* in /usr/share/doc/manual/apparmor-admin\_en/apparmor-admin\_en.pdf. Subprofiles are part of the profile for the application itself and are administered with the same tools (genprof, logprof).

---

## **Exercise 14-1    AppArmor**

In this exercise, you create, test, and improve a profile for the Firefox browser. This exercise has four parts.

In the first part create a profile for the Firefox browser. While using Firefox to generate log entries for the initial profile, just surf the web; do not access local files with Firefox.

In the second part, use Firefox to access a local file, such as `/usr/share/doc/packages/apparmor-docs/apparmor.7.html`. AppArmor should prevent you from viewing the file. Change the profile to allow access. You could use YaST, genprof, or complain and logprof for this purpose.

In the third part, install the Java browser plug-in package `java-1_4_2-sun-plugin`. Restart Firefox and use it to access a page containing a Java applet.

<http://java.sun.com/products/plugin/1.4/demos/plugin/applets.html> links to various demos. Firefox will not be able to show these. Change the profile again to be able to run Java applets. Use another method to do so than the method used in part two above.

In the fourth part, compare the profile you generated with those in `/etc/apparmor/profiles/extras/` for Firefox. Find out if your profile is more restrictive or more permissive compared with those profiles.

### **Detailed Steps to Complete this Exercise**

- Part I: Create a Profile for the Firefox Browser
- Part II: Modify the Profile for Firefox to Allow Read Access to the Local File System
- Part III: Use a Browser Plug-in
- Part IV: Compare the Profile You Created with Those From SLES 10

## Part I: Create a Profile for the Firefox Browser

Do the following:

1. Start **Yast**, enter the root password (**novell**).
2. Select **Novell AppArmor > Add Profile Wizard**.
3. At the prompt: **Application to Profile**, enter **firefox**.
4. Start Firefox by pressing **Alt+F2**, entering **firefox** and selecting **Run**. View some web pages. Close Firefox again.
5. In the YaST AppArmor Profile Wizard dialog, select **Scan system log for AppArmor events**.
6. Now you create the profile and need to answer several questions. Note that the application Firefox is quite complex and accesses several executables and files on the system.
  - a. Select **Inherit** for /bin/basename and other executed program .
  - b. For files and directories, choose an appropriate option, such as an #include, a filename, a directory, or a path with place holders, and select **Allow**.
7. When you are returned to the AppArmor Profile Wizard dialog, select **Finish**.
8. Make sure that the Firefox profile is in enforce mode by looking at **/sys/kernel/security/apparmor/profiles** using **cat**. There must be an entry **/usr/lib/firefox/firefox.sh (enforce)**. If it is not, execute **enforce firefox**.

## Part II: Modify the Profile for Firefox to Allow Read Access to the Local File System

Do the following:

1. Open a terminal window and `su -` to root (password **novell**).
2. Enter **tail -f /var/log/audit/audit.log**.
3. Start Firefox by pressing **Alt+F2**, entering **firefox** and selecting **Run**. Try to view the file `/usr/share/doc/packages/apparmor-docs/apparmor.7.html`. (You should not be able to do so.)
4. View the log file in the terminal window. You should see a reject message.
5. Stop viewing the log file by pressing **Ctrl+c**.
6. In the terminal window, enter **complain firefox**.
7. In Firefox, try again to access `/usr/share/doc/packages/apparmor-docs/apparmor.7.html`. You should now be able to access the file.
8. Start **Yast**, enter the root password (**novell**).
9. Select **Novell AppArmor > Update Profile Wizard**.
10. You are presented with the same interface as in Part I, where you can choose Allow, Inherit, Deny, etc. Make sure you are updating the Firefox profile, not some other profile.  
  
Sooner or later you should see an entry for the path `/usr/share/doc/packages/apparmor-docs/apparmor.7.html`. By selecting the Glob button three times, you can create a suggestion `/usr/share/doc/**`. Allow it by selecting Allow.
11. When all entries in the log file have been processed, select **Finish**.
12. Put the Firefox profile back in enforce mode by entering **enforce firefox** in the terminal window.



13. In Firefox, try to access files beneath /usr/share/doc/, like /usr/share/doc/packages/. You should be able to access them. However, accessing files elsewhere in the file system should not be possible.
14. Close Firefox.
15. Close YaST.

### Part III: Use a Browser Plug-in

Do the following:

1. Open a terminal window and **su -** to root (password **novell**).
2. Install the Java Browser Plug-in by entering (as root):  
**yast -i java-1\_4\_2-sun-plugin**  
Insert the appropriate media when prompted. Do not close the console window after the installation.
3. Start Firefox by pressing **Alt+F2**, entering **firefox** and selecting **Run**.
4. Visit <http://java.sun.com/products/plugin/1.4/demos/plugin/applets.html> and select one of the demos. (The demo should not work.)
5. In the console window, enter  
**genprof firefox**  
When asked to exercise the functionality of your application, select one of the demos again as in the previous step. The demo should work now.
6. Close Firefox.
7. Go back to the console window and press **s** to scan the logfile. Select **i** for inherit when the entry for java\_vm is shown.
8. Answer the subsequent questions with **Glob** and **Accept**, as applicable.
9. When all questions have been answered, press **f** to finish.

10. Start Firefox again and select another Java demo available at the URL given in Step 4. This should work now, despite the profile being in enforce mode again.

#### **Part IV: Compare the Profile You Created with Those From SLES 10**

Do the following:

1. Open a console window and view the profile  
`/etc/apparmor.d/usr.lib.firefox.firefox.sh` just created with `cat`.
2. Open another console window and view the profiles  
`/etc/apparmor/profiles/extras/usr.lib.firefox.firefox*`, using `cat`.
3. Compare the files. Note any differences and decide whether or not they are more restrictive than the one you created.

***(End of Exercise)***

## Objective 3      **Control AppArmor**

AppArmor can be controlled using the script **/etc/init.d/boot.apparmor** or the link to this script, **/sbin/rcapparmor**. This script takes the usual parameters start, stop, etc, but because AppArmor is not a daemon, their significance is slightly different.

To control AppArmor, you have to know how to

- Start and Stop AppArmor
- View AppArmor's Status
- Reload Profiles

### ***Start and Stop AppArmor***

To confine an application, AppArmor has to be active before the application starts. Therefore, AppArmor is usually activated early in the boot process.

If you do not want AppArmor to confine your applications any longer, use **rcapparmor stop**. This unloads the profiles, but the AppArmor kernel modules `apparmor` and `aamatch_pcre` remain loaded. **rcapparmor kill** unloads the kernel modules as well. In both cases, applications are no longer confined.

**rcapparmor start** activates AppArmor. However, only applications started after the activation of AppArmor are confined. Even if, for instance, a profile for Squid exists, Squid will not be confined if it was already running before you started AppArmor. To include Squid in AppArmor's protection, you need to restart Squid after activating AppArmor.

## ***View AppArmor's Status***

**rcapparmor status** gives you a general overview of profiles and processes:

```
da10:~ # rcapparmor status
apparmor module is loaded.
50 profiles are loaded.
49 profiles are in enforce mode.
1 profiles are in complain mode.
Out of 69 processes running:
5 processes have profiles defined.
5 processes have profiles in enforce mode.
0 processes have profiles in complain mode.
```

To emphasize the point that, after restarting AppArmor, processes need to be restarted to be again confined, have a look at the following:

```
da10:~ # rcapparmor stop
Unloading AppArmor profiles                                done
da10:~ # rcapparmor start
Loading AppArmor profiles                                  done
da10:~ # rcapparmor status
apparmor module is loaded.
50 profiles are loaded.
49 profiles are in enforce mode.
1 profiles are in complain mode.
Out of 62 processes running:
0 processes have profiles defined.
0 processes have profiles in enforce mode.
0 processes have profiles in complain mode.
```

Restarting one of the processes for which there is a profile changes the output of **rcapparmor status**:

```
da10:~ # rcnscd restart
Shutting down Name Service Cache Daemon      done
Starting Name Service Cache Daemon           done
da10:~ # rcapparmor status
apparmor module is loaded.
50 profiles are loaded.
49 profiles are in enforce mode.
1 profiles are in complain mode.
Out of 62 processes running:
1 processes have profiles defined.
1 processes have profiles in enforce mode.
0 processes have profiles in complain mode.
```

The output of AppArmor does not contain specific data regarding the profiles or the processes being confined.

A list of the profiles loaded is kept in **/sys/kernel/security/apparmor/profiles**. It might look like the following:

```
da10:~ # cat /sys/kernel/security/apparmor/profiles
/usr/sbin/traceroute (enforce)
/usr/sbin/squid (enforce)
/usr/sbin/sendmail (enforce)
/usr/sbin/postqueue (enforce)
...
/usr/lib/postfix/bounce (enforce)
/usr/lib/firefox/firefox.sh (complain)
/usr/bin/ldd (enforce)
...
```

The command **unconfined** lists processes that have bound sockets but have no profiles loaded:

```
da10:~ # unconfined
2659 /sbin/portmap not confined
2659 /sbin/portmap not confined
2694 /usr/lib/zmd/zmd-bin not confined
2756 /usr/sbin/slpd not confined
2756 /usr/sbin/slpd not confined
2756 /usr/sbin/slpd not confined
2756 /usr/sbin/slpd not confined
2756 /usr/sbin/slpd not confined
2756 /usr/sbin/slpd not confined
2831 /usr/sbin/cupsd not confined
2831 /usr/sbin/cupsd not confined
2874 /usr/sbin/sshd not confined
2905 /usr/sbin/sshd not confined
2905 /usr/sbin/sshd not confined
3040 /usr/lib/postfix/master not confined
3040 /usr/lib/postfix/master not confined
```

This does not give information about processes with profiles that are not confined because they were running already when AppArmor was activated. To spot those, you would have to compare the output from `ps` with the content of `/sys/kernel/security/profiles` and restart any processes that should be confined.

## ***Reload Profiles***

If you have changed profiles in `/etc/apparmor.d/` manually with an editor (not by using the AppArmor tools like `logprof`), you have to reload the profile or profiles concerned. The command to use is **`rcapparmor reload`**. **`rcapparmor restart`** is equivalent to reload; it does not stop and then start AppArmor, but it does reload the profiles. Processes that were confined before **`rcapparmor reload`** was issued remain confined (unless you deleted their profile or changed their status from `enforce` to `complain`).

The commands **`enforce`** and **`complain`** toggle the status from `enforce` to `complain` and vice versa, and reload the profiles concerned.

## Objective 4 Monitor AppArmor

There are two ways to monitor AppArmor:

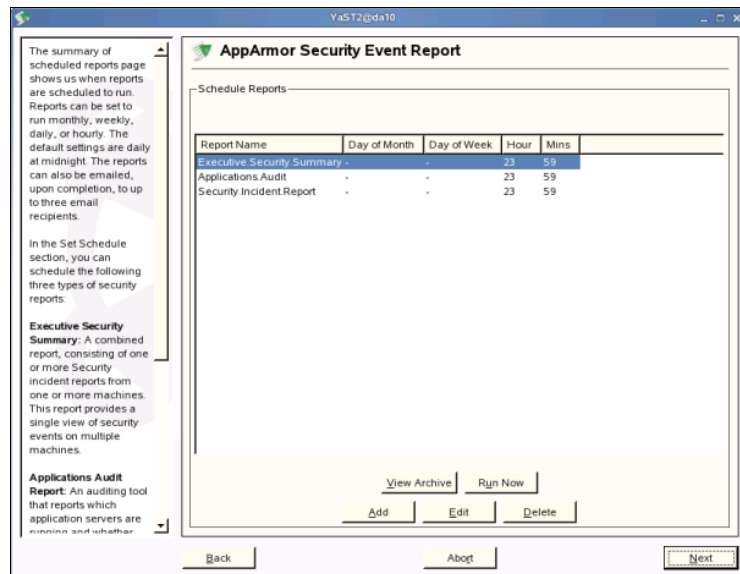
- Security Event Report
- Security Event Notification

### Security Event Report

The YaST module to configure and view AppArmor security event reports can be launched by starting YaST and selecting **Novell AppArmor > AppArmor Reports**. It can also be launched directly from a console window as root by entering **yast2 SD\_Report**.

The dialog that opens up shows when security event reports are generated:

Figure 14-7



By default these are created once a day at midnight.



Using the buttons **Add**, **Edit**, or **Delete**, you can schedule new security incident reports, edit the existing ones, for instance, to set the email address that should receive the report, or delete event reports.

Selecting a report and then selecting **Run Now** either shows the result directly, or, in the case of the Security Incident Report, first opens a dialog where you can fine tune the content of the resulting report:

**Figure 14-8**

The Report Configuration dialog enables you to filter the report selected in the previous screen. To filter by **Date Range**:

1. Click **Filter By Date Range**. The fields become active.
2. Enter the start and end dates that delineate the scope of the report.
3. Enter other filtering parameters. See below for definitions of parameters.

The following definitions help you to enter the filtering parameters in the Report Configuration Dialog: **Program Name Pattern**: When you enter a program name or pattern that matches the name of the executable process of interest, the report will display security events

**Report Configuration Dialog**

☐ **Filter By Date Range**

Select Date Range

Enter Starting Date/Time

Hours: [0] Minutes: [0] Day: [1] Month: [1] Year: [2005]

Enter Ending Date

Hours: [0] Minutes: [0] Day: [1] Month: [1] Year: [2005]

Program name: Profile name: PID number: Severity: [All]

Detail: Access Type: [R] Mode: [All]

Export Type: [None] Location to store log: [/varlog/apparmor/reports-exported] [Browse]

Back Abort Next

The help text on the left explains the available options. Once you configured what you want to have included in your report, select **Next**. The report is displayed, showing the security events:

**Figure 14-9**

**Security Incident Report (SIR):** A report that displays security events of interest to an administrator. The SIR reports policy violations for locally confined applications during the specified time period. The SIR reports policy exceptions and policy engine state changes. These two types of security events are defined as follows:

- **Policy Exceptions:** When an application requests a resource that's not defined within its profile, a security event is generated.
- **Policy Engine State Changes:** Enforces policy for applications and maintains its own state, including when engines start or stop, when a policy is

**AppArmor On-Demand Report**

On Demand Event Report - Page 1 of 1

Host	Date	Program	Profile	PID	Severity	Mode	Detail
da10	2006-07-05 15:03:03	firefox-bin	/usr/lib/firefox/firefox.sh	9026	1	r	/usr/share/doc/p...
da10	2006-07-05 15:07:09	firefox-bin	/usr/lib/firefox/firefox.sh	9114	1	r	/usr/share/doc/p...
da10	2006-07-05 15:08:35	firefox-bin	/usr/lib/firefox/firefox.sh	9114	1	r	/usr/share/doc...
da10	2006-07-05 15:09:08	firefox-bin	/usr/lib/firefox/firefox.sh	9114	1	r	/usr/share/doc...
da10	2006-07-05 15:11:19	firefox-bin	/usr/lib/firefox/firefox.sh	9114	U	r	/etc
da10	2006-07-05 15:16:29	firefox-bin	/usr/lib/firefox/firefox.sh	9905	4	r	/home/geeko/foi
da10	2006-07-05 15:16:29	firefox-bin	/usr/lib/firefox/firefox.sh	9905	4	r	/home/geeko/foi
da10	2006-07-05 15:16:32	firefox-bin	/usr/lib/firefox/firefox.sh	9905	5	r	/usr/lib/jvm/java-
da10	2006-07-05 15:16:32	firefox-bin	/usr/lib/firefox/firefox.sh	9905	5	r	/usr/lib/jvm/java-
da10	2006-07-05 15:17:45	firefox-bin	/usr/lib/firefox/firefox.sh	9905	5	r	/usr/lib/jvm/java-
da10	2006-07-05 15:17:45	firefox-bin	/usr/lib/firefox/firefox.sh	9905	5	r	/usr/lib/jvm/java-
da10	2006-07-05 15:19:06	firefox-bin	/usr/lib/firefox/firefox.sh	9969	4	r	/home/geeko/foi
da10	2006-07-05 15:19:06	firefox-bin	/usr/lib/firefox/firefox.sh	9969	4	r	/home/geeko/foi
da10	2006-07-05 15:19:19	firefox-bin	/usr/lib/firefox/firefox.sh	9969	5	r	/usr/lib/jvm/java-
da10	2006-07-05 15:19:19	firefox-bin	/usr/lib/firefox/firefox.sh	9969	5	r	/usr/lib/jvm/java-
da10	2006-07-05 15:19:19	firefox-bin	/usr/lib/firefox/firefox.sh	9969	5	r	/usr/lib/jvm/java-
da10	2006-07-05 15:19:19	firefox-bin	/usr/lib/firefox/firefox.sh	9969	5	r	/usr/lib/jvm/java-
da10	2006-07-05 15:19:19	firefox-bin	/usr/lib/firefox/firefox.sh	9969	5	r	/usr/lib/jvm/java-
da10	2006-07-05 15:19:19	firefox-bin	/usr/lib/firefox/firefox.sh	9969	5	r	/usr/lib/jvm/java-
da10	2006-07-05 15:19:19	firefox-bin	/usr/lib/firefox/firefox.sh	9969	5	r	/usr/lib/jvm/java-
da10	2006-07-05 15:23:17	firefox-bin	/usr/lib/firefox/firefox.sh	10025	4	r	/home/geeko/foi
da10	2006-07-05 15:23:17	firefox-bin	/usr/lib/firefox/firefox.sh	10025	4	r	/home/geeko/foi

First Page Previous Sort Forward Last Page Go to Page

Back Abort Done

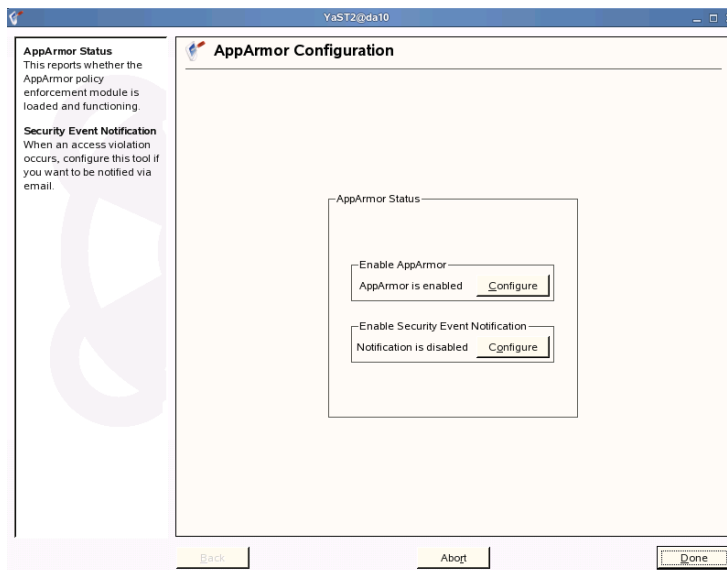
Depending on your configuration in the previous dialog, it is also saved to disk, by default in the directory `/var/log/apparmor/reports-exported/`.

## Security Event Notification

To configure the security event notification, start **YaST** and select **Novell AppArmor > AppArmor Control Panel**, or start the module directly from a console window as root by entering **yast2 subdomain**.

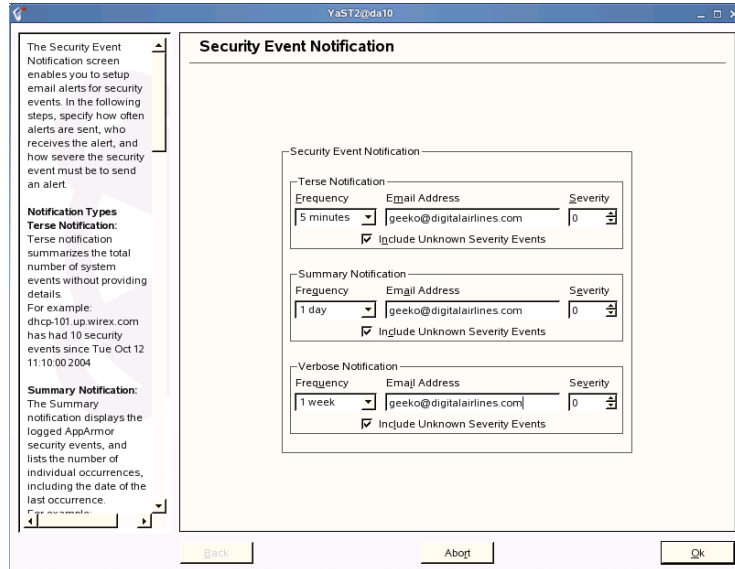
The following dialog opens up:

**Figure 14-10**



Select **Configure** in the **Enable Security Event Notification** box. A dialog opens up where you can configure the frequency of the notifications, email addresses, and the severity levels the reports should cover:

**Figure 14-11**



Select **Ok** to save your configuration. Close the AppArmor configuration window by selecting **Done**.

## Summary

Objective	Summary
1. Improve Application Security with AppArmor	<p>AppArmor imposes limits on processes based on rules set in profiles.</p> <p>Even if compromised, programs are limited in what they are allowed to read, write, and execute. This even holds true for programs running with root permissions.</p>
2. Create and Manage AppArmor Profiles	<p>There are YaST modules as well as command line tools to create and manage AppArmor profiles.</p> <p>The main YaST modules are the Add Profile Wizard, and the Update Profile Wizard.</p> <p>The corresponding command line tools are autodep, genprof, and logprof.</p> <p>Profiles are text files located in <code>/etc/apparmor.d/</code> and can be maintained with any text editor.</p>

Objective	Summary
3. Control AppArmor	<p>AppArmor is started early during the boot process.</p> <p><b>/etc/init.d/boot.apparmor</b> is the script used to control AppArmor. <b>/sbin/rcapparmor</b> is a link to it.</p> <p>The parameters <b>stop</b> and <b>kill</b> end the confinement of processes by AppArmor. <b>start</b> is used to activate confinement of processes—but only processes with profiles that are started after AppArmor has been activated are confined. <b>reload</b> or <b>restart</b> reload the profiles.</p> <p><b>enforce</b> and <b>complain</b> toggle enforce and complain mode.</p> <p><b>unconfined</b> lists processes with bound sockets that have no profile.</p>
4. Monitor AppArmor	<p>AppArmor can be configured via YaST to create reports on security events and to send email messages to inform on security events.</p>

# Index

## Symbols

#include 14-4  
 /etc/apparmor.d/ 14-2, 14-8  
 /etc/apparmor.d/abstractions/ 14-6  
 /etc/apparmor.d/tunables/ 14-5  
 /etc/apparmor/abstractions/ 14-4  
 /etc/apparmor/profiles/extras/ 14-8  
 /etc/default/passwd 6-18  
 /etc/init.d/boot.apparmor 14-3, 14-27  
 /etc/inittab 6-20  
 /etc/login.defs 6-18  
 /etc/pam.d/ 6-4  
 /etc/permissions\* 6-25  
 /etc/permissions.d/ 6-25  
 /etc/permissions.local 6-25  
 /etc/permissions.paranoid 6-25  
 /etc/permissions.secure 6-25  
 /etc/security/pam\_pwcheck.conf 6-18  
 /etc/sysconfig/syslog 7-3  
 /etc/syslog-ng/syslog-ng.conf 7-4, 13-8  
 /etc/syslog-ng/syslog-ng.conf.in 7-4, 13-8  
 /sbin/rcapparmor 14-27  
 /sys/kernel/security/apparmor/profiles 14-29  
 /var/log/audit/audit.log 14-10  
 /var/log/wtmp 6-23

## A

aamatch\_pcre kernel module 14-2

Add Profile Wizard 14-15  
 Allow 14-12  
 AppArmor 14-1  
 apparmor kernel module 14-2  
 AppArmor profiles 14-4–14-5  
 AppArmor profiles, administration 14-8  
 AppArmor profiles, permissions 14-6  
 AppArmor rules 14-5  
 AppArmor, monitoring 14-32  
 AppArmor, starting and stopping 14-27  
 AppArmor, status 14-28  
 application browser 2-8  
 autodep 14-15, 14-17

## B

background 3-1  
 boot settings 6-20  
 booting 2-2  
 bottom panel 2-4, 2-13

## C

Capability, POSIX 14-5  
 class 3-15, 3-32  
 clock 2-6  
 CLP Intro-3, Intro-5  
 complain 14-15  
 component 3-2, 3-7, 3-30, 11-27  
 configuration 3-9, 3-11–3-12, 3-23–3-26,

3-31, 11-24, 11-27–11-28, 11-52,  
12-10

configure 11-1, 11-27, 11-52, 12-1

controller 3-23

create 3-9, 3-12, 11-27, 11-52, 12-10

Ctrl+Alt+Del 6-20

## D

date 2-6

dd 4-3

delete a profile 14-17

Deny 14-12

Destination (syslog-ng.conf) 7-9

device 2-11, 3-15

directory 3-4, 3-7–3-8, 3-14–3-16, 3-24,  
3-31, 8-8, 11-24, 11-29

DNS 9-7

## E

Edit 14-13

emblem 2-16

encrypted 11-27

enforce 14-17

## F

facility (syslog) 7-5

file

system 3-6, 3-14–3-15, 3-32

file manager 2-15

filters (syslog-ng.conf) 7-8

find 2-17

folder 2-11

## G

genprof 14-18

Glob 14-12

Glob w/Ext. 14-13

GNOME 2-1, 2-3, 2-5–2-6, 2-10, 2-19

graphical user interface 2-12

Group ID Settings 6-24

## H

hardware 3-1–3-3, 3-6–3-7, 3-9, 3-11, 3-14,  
3-17, 3-24, 3-30–3-31

header 12-10

home directory 2-15

## I

icon 2-10, 2-12–2-13, 2-16

Inherit 14-11

## J

John (password cracker) 6-11

## K

KDE 2-5

## L

language 2-5

launcher 2-11

LDIF 11-51

link 2-11

LOAD 3-6, 3-12, 3-17, 3-30

locate 6-25

locatedb 6-25



Log Path (syslog-ng.conf) 7-9  
logging to remote host 13-8  
loghost 13-8  
login 2-3  
login dialog 2-2  
login settings 6-22  
logprof 14-20  
lvcreate 4-5, 4-7  
lvextend 4-5, 4-7  
lvreduce 4-5, 4-7  
lvscan 4-5, 4-7

## M

Magic SysRq Keys 6-26  
management 3-1  
mandatory access control 14-1  
Manually Add Profile 14-14  
master 12-11  
memory 3-6, 3-12, 3-15

## N

Nautilus 2-15  
NetworkManager 5-2  
New Profile Wizard 14-8  
nm-applet 5-2  
nm-tools 5-2  
Novell AppArmor 14-1  
Novell Customer Center Intro-6

## P

PAM 6-2  
PAM, arguments 6-8  
PAM, Configuration Files 6-4

PAM, control flags 6-6  
PAM, documentation 6-12  
PAM, Illustration 6-3  
PAM, module types 6-5  
PAM, modules 6-7  
panel 2-4, 2-6, 2-13–2-14  
partprobe 4-2  
password 2-3  
password security settings, exercise 6-27  
password settings 6-18  
Password, secure 6-11  
physical 3-2, 3-14, 3-30  
Pluggable Authentication Modules 6-2  
POSIX capabilities 14-5  
power management 2-6  
printer 3-2  
priority (syslog) 7-6  
Profile 14-11  
profile, delete 14-17  
profile, new 14-8  
profile, update 14-14  
profiles, reload 14-31  
pvcreate 4-3, 4-7  
pvmove 4-4, 4-7  
pvscan 4-4, 4-7

## R

reboot 2-5  
recent match 13-7  
root 2-14, 3-12, 12-10

## S

SCSI 3-6, 3-15  
security 11-27

- Security Settings 6-16
- security updates 13-2
- security, application 14-2
- server 3-6, 3-31, 11-1–11-2, 11-6,  
11-27–11-29, 11-52, 12-1
- session 2-5
- shutdown 2-4–2-5
- Side Panel 2-16
- size 3-8
- SMTP 12-11
- software 3-2, 3-30, 11-1–11-2, 11-6, 11-21,  
11-52, 12-24, 12-40
- source (syslog-ng.conf) 7-7
- start 3-6, 3-9, 3-23, 3-31, 11-2, 11-28, 12-11
- state 3-14
- storage 3-4
- subdirectory 3-15
- Subdomain 14-3
- SuSEconfig 13-8
- SYS 3-14–3-15, 3-32
- syslog-ng 13-8
- SysRq 6-26
- system 3-1, 3-6–3-7, 3-9, 3-14–3-15, 3-25,  
3-31–3-32

## T

- task manager 2-6
- terminal 2-12, 2-19
- terms 3-2
- time 2-6, 3-25–3-26
- type 12-43

## U

- unconfined 14-11, 14-30
- update 2-6

- Update Profile Wizard 14-15
- updatedb 6-25
- user 2-14–2-15
- User Access 6-1
- User ID settings 6-23

## V

- vgcreate 4-4, 4-7
- vgexpand 4-4, 4-7
- vgreduce 4-4, 4-7
- vgremove 4-4, 4-7
- vim 14-21
- virtual terminal 2-19
- volume control 2-6

## W

- window manager 2-5